MEMORANDUM
RM-5563-PR
JANUARY 1968

PLANET: PART V– REPORTS AND ANALYSIS LIBRARY

B. J. Voosen, S. Glaseman, R. J. Young and Judy Judd

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM

RM-4663-PR

JANUARY 1969

PLANET: PART V—REPORTS AND ANALYSIS LIBRARY

B. J. Voosen, S. Glaseman, R. J. Young and Judy Judd

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

The RAND Corporation

Published by The RAND Corporation

PREFACE AND SUMMARY

PLANET (Planned Logistics Analysis and Evaluation Technique) is a series of four computer simulation models designed to examine the hardware-configuration/operations/logistics support interactions of a variety of weapon systems in a single or multibase environment. Its purpose is to help the manager gain an understanding of the operation of his system and find a rationale for allocating resources effectively and efficiently.

The PLANET complex comprises five computer programs:

1) The Availability and Base Cadre Simulator (ABC) furnishes the framework for the logistics resources assigned to a support base or bases.

2) The Bench Repair Simulator (BR) processes the reparables through the base repair shops or diverts them to a depot, thus converting the reparables to serviceables.

3) The Depot Transportation Simulator (DT) processes the movement of reparables from the base(s) to the depot(s) or factory and return.

4) The Depot Repair and Overhaul Simulator (DR&O) simulates the functions in a repair or overhaul facility.

5) The Reports and Analysis Library consists of twelve different output programs.

The simulators can be used separately to examine specific areas of the logistics system, or conjointly to simulate the complete weapon-system operation from the site or point of demand through to the depot.

The Reports and Analysis Library described here contains the twelve report programs with operating instructions. All reports are designed for use by managerial personnel. The manager may select from the library those programs best suited for analysis of his particular problem. Even though the output programs cover a wide spectrum of problem areas, it can be anticipated that additional outputs will in some cases be required. Either the output programs can be modified to incorporate any additional data required, or new programs can be written with relative ease.

Although PLANET is designed as an advanced planning tool, it can be used to assess periodically whether the logistics support planned will maintain a system or equipment effectively and economically. These assessments combine the relevant reliability, maintainability, and performance parameters for the weapon system. This enables the systematic development of an integrated logistics support plan for systems and equipment at all levels of maintenance for its programmed life cycle.

While PLANET is programmed in SIMSCRIPT, the user need not be a skilled SIMSCRIPT programmer to conduct a simulation. We have included the necessary step-by-step instructions as well as the necessary SIMSCRIPT instructions to permit managers to assemble the data in a form acceptable to the models.

FOREWORD

In general, computer simulation is a way of using a computer to produce a reasonable likeness of the behavior of a system under study. Simulation models are only representations of reality. Of necessity, the likeness of the system under study is "scaled down" to manageable size for the computer. Simulation models, therefore, are based on the designer's concept of what the key elements of the system are, and how they operate and interact on the system.

This being so, one cannot say a priori that one model is better than another. A manager should always strive for the lowest-cost model that suits his purpose. Since computer simulation models generally are explanatory, the analyst must first determine if a particular model sufficiently represents his system. In short, the analyst must first understand the model and then strike a compromise between realism and simplicity.

The size and complexity of the problem that the system manager would like to examine increase as a function of the interrelationships to be considered. It is very difficult, for example, to visualize the impact of a change in sortie rate on the personnel requirements in a depot overhaul facility, or even the effect of a change in reliability/maintainability parameters on the operational capability if support shortages exist at the higher echelons of maintenance. In short, while problems can be bounded and scaled down to manageable size, it is often desirable to view the analysis in a broader context to observe more of the interactions.

PLANET was developed as a logistics prediction and estimating tool. Its purpose is to help the manager gain an understanding of the operation of his system and find a rationale for allocating resources efficiently. Real world observations of a system help serve the same end, but simulations permit more varied, controlled and complete ranges of experience, usually at far less cost and much earlier in the life of the weapon system.

Coincident with the PLANET development program, DOD Directive 4100.35 dated June 19, 1964 was issued. The following is quoted from

that directive:

> The primary objective of this Directive is to assure that the development of effective logistic support for systems and equipments is systematically planned, acquired, and managed as an integrated whole (by interlocking the elements of logistic support) to obtain maximum material readiness and optimum cost effectiveness.
>
> Integrated Logistic Support - Integrated Logistic support is a composite of the elements necessary to assure the effective and economical support of a system or equipment at all levels of maintenance for its programmed life cycle. It is characterized by the harmony and coherence obtained between each of its elements and levels of maintenance.

We believe that simulation models such as PLANET can be used to develop an "Integrated Logistics Support" plan for a spectrum of weapon systems.

The history of the development of PLANET might be of interest. Prior to the actual coding of the programs, approximately one man-year was devoted to the problem of how best to structure the models. It was obvious that the bulk of the computers envisioned by the projected release date (1966) would have memories of 32K words. Although larger computers were being proposed at the time, we had no guarantee that very large computers (greater than 96K) would be readily available to prospective users. Therefore the problem of how to structure the programs to be useful, regardless of computer size, had to be faced.

In addition, we wanted to structure the simulation programs so that the internal logic of the simulator could be modified for special applications with relative ease. This required that the family of weapon systems that the model is designed to imitate be as broad as possible while the program itself be segmented into as many small subroutines as practicable.

The result of the planning phase was that the simulators would be bounded in a logical order of Flightline or site, Base, Depot and a link between Base and Depot. Each segment of the total simulation package must be capable of being used on a machine with the limited capacity of 32K. In some instances, this constraint limited the amount of detail we would have liked to include. In addition, since there

appears to be a trend toward the procurement of larger (larger than 32K) machines, the models should be easily assembled into larger, more detailed simulation programs.

After many months of examining logistic systems and plans, a commonality among them appeared to emerge that indicated the feasibility of such an undertaking. Although in many instances (particularly in the comparison of aircraft and missile logistic systems) the jargon used to describe specific functions was completely different, the functions to be performed were similar. Even though the operating parameters for the simulation would be different, this meant that the same computer program logic could be used if the real world jargon could be defined into common terms.

We therefore have tried to use, as much as possible, the functional description of the various logistics actions and activities, and hope that users will be able to translate the jargon into functional terms for use in the simulation.

Coding of the programs began in January 1964. By October 1964 the first (ABC simulator) of the four simulators was available for debugging and proof testing.

For each simulator, debugging consists of tracing (using trace routines coded into the program) each event through its cycle during a simulation run.

The proof testing consisted of inputting a set of empirical data, computing each value that was to be generated by the computer by manual or analytical methods, then comparing the empirical inputs with the actual experience that occurred in the real world. This has been done for both missile and aircraft data sets.

By March 1967, all models were coded and proof testing of the last was nearing completion. There remained only the marriage of the four programs to ensure that they would in fact work together.

In total approximately nine man-years of coding effort was used to code, debug and proof test the models, and approximately 200 hours of computer operating time (this is in addition to the one man-year previously mentioned). It is easy to see, therefore, why some organizations would have difficulty in developing models of this size and complexity.

In conclusion, we hasten to point out that while PLANET is developed as a "general purpose simulation model," it is not a panacea. It is limited in purpose and scope. However, we have endeavored to structure the simulators so that as the need arises additional complexity can be added and the models thus can be expanded.

ACKNOWLEDGMENTS

To acknowledge everyone who contributed to PLANET would require many pages. We would feel negligent, however, if we did not acknowledge the efforts of all the programmers who developed the many programs contained in this Memorandum.

1. Miss Pat Love developed the Cost Effectiveness program (number 8).

2. Mr. Richard Villanueva developed the Bench Repair Capability and NRTS programs (numbers 9 and 10).

3. Mr. Al Nelson developed the Aircraft Recovery Package Report Generator (numbers 4 to 7).

4. Mr. Steven Glaseman developed the Depot Transportation output program (number 11).

5. Last but not least, Mrs. Shirley Ballinger developed the Depot Repair and Overhaul Capability output program and has been of great assistance during the early production runs of PLANET.

CONTENTS

CONTENTS (Continued)

Part 1

INTRODUCTION AND INITIALIZATION INSTRUCTIONS

I. INTRODUCTION

In the research phase of weapon system development, the system manager faces the problem of designing a weapon system to meet specified operational objectives at a minimum cost. If he considers only the operational environment, the resulting hardware may be very difficult and costly to support. It often becomes apparent later, during development, that if some particular factor had been considered earlier, a more effective system could have resulted for the same cost.

Cost/effectiveness analysis, if properly used, brings into focus the parameters that affect mission capability. The object is usually to minimize the cost at which a specified level of effectiveness can be maintained; this involves a comparison of alternative ways of designing and supporting a particular system for a given mission.

The operating procedure for PLANET consists of a two-phase operation: first, the Simulation phase (S phase); second, the Report phase (R phase). The simulators can be used singly or assembled in various configurations to represent a more detailed description of the logistics system to be examined. Regardless, the output from the S phase will be a tape listing of selected variables accumulated during the simulation. This tape(s) can be retained as a permanent record of the simulation. From this tape(s), the desired reports are generated by using the following library of programs. There are twelve programs to choose from.

NOR TIME SUMMARY

The NOR (not operationally ready) time summary is a listing of the ABC Simulator NOR time distributions and a count of the unscheduled demands for the simulated fleet.

WEAPON SYSTEM AVAILABILITY

The weapon system availability program is designed for use with missile simulations. From the ABC output tape, it displays the missile off-alert time by tail number as well as a chronological, time-oriented

listing of what happened while the missile was off alert. Details include the time a team was dispatched from the support base, the arrival time at the site, when the maintenance action was completed, and when the missile was returned to alert status.

LOGISTICS RESOURCE UTILIZATION

The ABC logistics information is presented in three parts: Spares, Personnel, and Equipment. The spare part data presented by this report contain information regarding stock levels, NORS (not operationally ready--supply) time, NORS count, and demand quantities. The personnel report contains the man-hour accounting information. This report displays the utilization factors for each personnel type and the man-hours consumed by various tasks. The equipment data presented contain information regarding the utilization of maintenance equipment and facilities. NORE (not operationally ready--equipment) time, NORE count, and demand quantities are included.

The logistics resource utilization report can be used for either aircraft or missile simulations.

AIRCRAFT RECOVERY PACKAGE

The following four programs are peculiar to aircraft simulations; they cannot be used with in-place missile simulations.

Aircraft Recovery Time Distributions

The aircraft recovery histogram presents a display of the entire aircraft (as opposed to system or subsystem) recovery. The display is divided into two halves: one half shows the touchdown time by workshift, the corresponding second half shows the type of sortie. Below this display are a number of statistical computations for facilitating analysis, e.g., the average recovery time (for unscheduled maintenance) and the operationally ready (OR) time lost in recovering aircraft from the effects of the sorties, etc.

System Recovery

The aircraft System Recovery program produces a summary of the actions to clear unscheduled maintenance demands. This display serves two purposes: it enables the monitoring of break and recovery rates, and it provides a set of job standards for unscheduled maintenance. Below the display are a number of statistical computations for facilitating analysis, e.g., the total OR time lost to this subsystem, the average number of men working on system recovery, etc.

Work Center Recovery

The Work Center Recovery output is a series of frequency distributions (one for each hour of the simulated day), showing how the flight-line demands were distributed throughout the 24-hour period. The purpose of this display is to aid in determining shift assignments for personnel.

The work center data the recovery program produces are: the touchdown time, the time the work center began the first job and ended the last job, the number of people at work in each 30-minute trial period, and the conventional AFM 66-1 data.

Failure List

The Failure List provides a graphical history of break-rate information. This display presents the flight-line demand data. The break-rates are inferred from the demands, both scheduled and unscheduled, against each individual unit. The probability data are computed only for 25 or more sorties; the mean sorties-to-failure data are computed if five or more fixes have occurred.

COST/EFFECTIVENESS PROGRAM

The Cost/Effectiveness program can be set to examine two cost factors: the total system cost and the logistics support costs. Logistics costs are the summation of the various resource and facility costs specified for the simulation. Total system costs are the logistics

costs just mentioned plus the cost of the items being simulated (sites). The measure of effectiveness is operationally ready (OR) time.

BASE SHOPS MAINTENANCE CAPABILITY

As the title implies, this output program is used to display the outputs from the Bench Repair Simulator. The output display consists of five parts: the input to each shop and its output and repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare-part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of the simulation.

NRTS PROGRAM

The NRTS (not reparable this station) data display shows the reparables shipped off base for repair. It displays the pipeline time for the reparables. This program is used primarily as an input to the Depot Transportation Simulator when the simulators are operated separately.

TRANSPORTATION CAPABILITY

This output program is the output display for the Depot Transportation Simulator. The outputs are presented in six parts: the tonnage delivered to the various bases for each type of priority cargo; the quantity of each cargo type delivered to the various delivery points; the mileage and in-transit time for each transportation mode; the utilization rates of the vehicles assigned to the transportation system; cargo processing time and quantities; and queueing factors associated with cargo transportation at each base.

DEPOT CAPABILITY

The Depot Capability output program is used with the Depot Repair and Overhaul Simulator. The output displays consist of six parts:

the input to the depot, and the depot's output and in-process time distribution; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare-part type; stock levels, spare-part repair time, stockouts, and demands for each spare part; detailed information for each activity about its performance during each simulated period; and queueing factors and downtime for any "special" type of activity within the repair or overhaul process.

As previously mentioned, the manager may select from this library of programs those best suited for analysis of his particular problem. Even though the output programs cover a wide spectrum of problem areas, it can be anticipated that additional outputs will be required for some analyses. Either the output programs can be modified to incorporate any additional data required, or new programs can be written with relative ease.

This Memorandum is divided into two parts. Part 1 contains the introduction, which includes a brief description of each of the reports. Section II presents the SIMSCRIPT instructions required to initialize any of the report programs.

Part 2 is the library of programs, including a description of each program, the initialization requirements, a program description oriented to the skilled SIMSCRIPT programmer who may wish to make a change, and a listing of the SIMSCRIPT SØURCE program.

II. INITIALIZATION

All of the Report programs are written in SIMSCRIPT as Non-simulation programs. Non-simulation programs, as with the PLANET simulation programs (Refs. 2, 3, 4, 5), are translated by SIMSCRIPT into FØRTRAN source programs, which are then compiled by the FØRTRAN Monitor into a FØRTRAN object program. Once the program has been compiled, the composition of the execute deck is as shown in Fig. 1.

The execution of object programs is accomplished in the usual FØRTRAN manner, and whatever control cards may be required by a particular installation appear first.

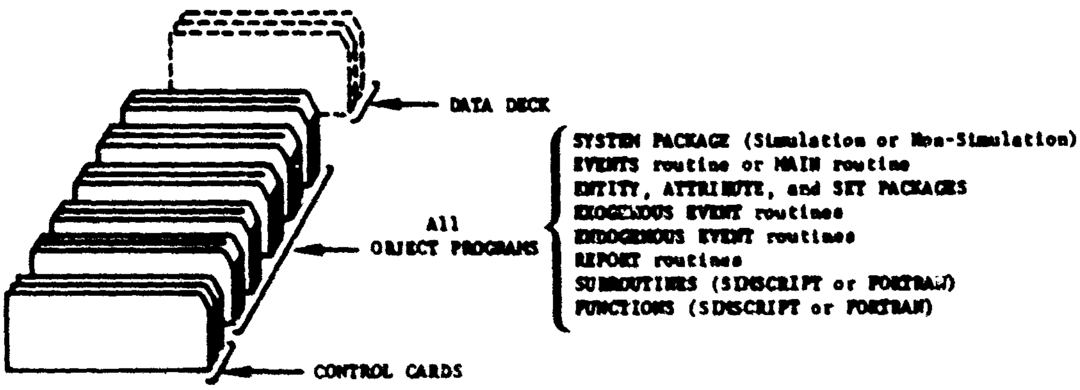


Fig. 1 -- Execute Deck

The object programs may appear in any order; they are:

A Simulation Package
An Events Routine or a Main Routine
All Entity, Attribute, and Set Packages
All Exogenous Events Routines
All Report Routines
All Subroutines (SIMSCRIPT or FØRTRAN)
All Functions (SIMSCRIPT or FØRTRAN)

The object programs are constructed from the information contained in the appropriate Report program. We shall assume at this point that the analyst has an "Object Deck" available and is now ready to assemble a "Data Deck."

DATA DECK

The composition of the Data Deck is shown in Fig. 2. The various elements and the data requirements for each are discussed below.

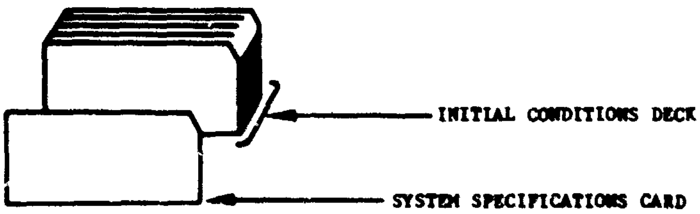


Fig. 2 -- Data Deck

System Specification Card

The first card in the Data Deck is the System Specification Card. In Col. 1 must be the number 1. In Cols. 7 through 12 is punched the maximum "Array Number" as in Fig. 3. Only these two entries are required. For a complete System Specification Card format, refer to the top of Fig. 4.

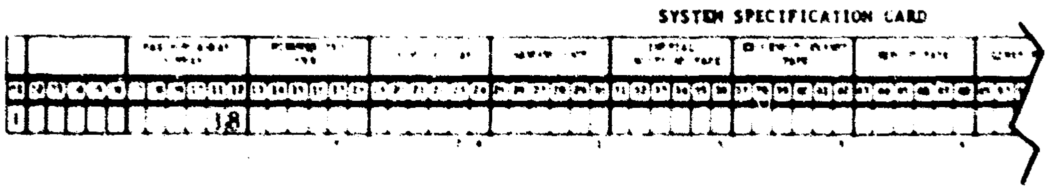


Fig. 3 -- System Specification Card

Initial Conditions Deck

The Initial Conditions Deck consists of all Initialization Cards and Data Cards. Before discussing each of the cards contained in the Initial Conditions Deck, let us first discuss the SIMSCRIPT Initialization Form, because the Initial Conditions Deck is created from the information contained in the Initialization Form.

The specification of initial conditions for the Report Generators is a very simple process. Only a few arrays need be initialized in all cases. The appropriate arrays and the required values (data) are specified for each report.

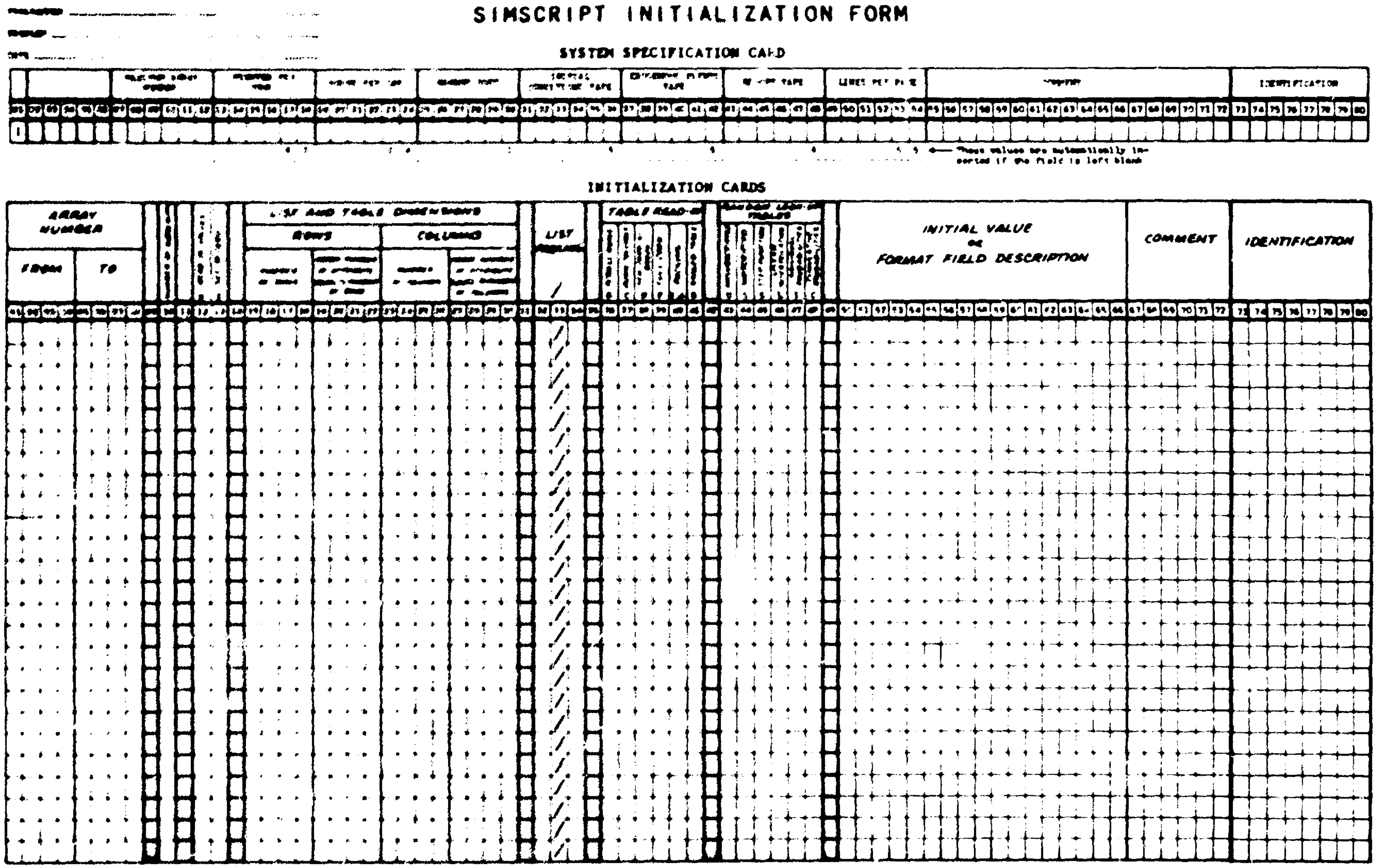


Fig. 4 -- Initialization Form

The Initialization Deck consists of Data Cards plus Initialization Cards punched from the Initialization Form shown in Fig. 4. Every Array Number from "1" up to the largest must be considered in sequential order in the Initialization Cards. The complete sequence of Array Numbers must be accounted for.

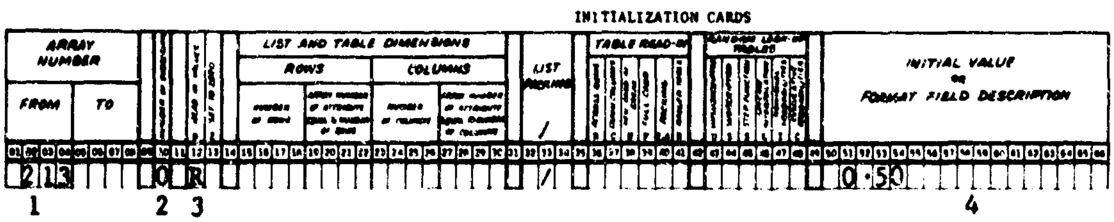
Procedures for preparing the Initial Conditions Deck are discussed under the following headings:

Unsubscripted Permanent Attributes
Single-subscripted Permanent Attributes
Double-subscripted Permanent Attributes

Unsubscripted Permanent Attributes. Each unsubscripted Permanent Attribute (System Attribute) must have its initial value read in or set equal to zero.

Initial values of unsubscripted Permanent Attributes may be separately specified by means of individual Initialization Cards. They may also be handled in groups by means of a single Initialization Card followed by Data Cards. To be initialized as a group, the System Attributes in the group must have consecutive Array Numbers. Their values must also be read in by using the same FORMAT statement Field Description.

Figure 5 shows the entries required to read in the initial value of a single System Attribute. The initial value can be set to zero by



1. Enter the Array Number in Cols. 1 through 4. The unit's position of the Array Number must be in Col. 4.
2. Enter a zero in Col. 10.
3. Enter an "R" in Col. 12.
4. Enter the Initial Value as an integer or decimal number anywhere in Cols. 50 through 66. Formats other than integer or decimal (e.g., hours or alphanumeric) must be read from the Data Cards.

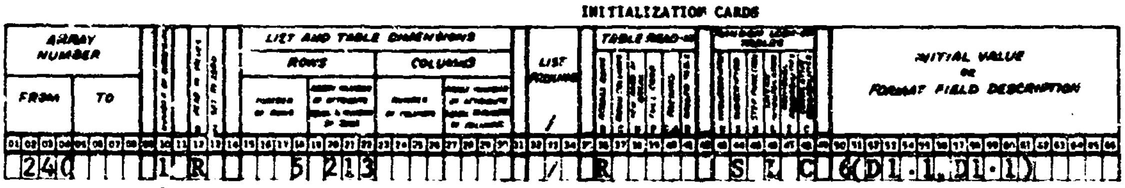
Fig. 5 -- Initialization Card Entries for a Single Unsubscripted Permanent Attribute

inserting a zero in Cols. 50 through 66, or by leaving Col. 12 blank and inserting a "Z" in Col. 13.

Single-subscripted Permanent Attributes. If the initial values are to be read in, a separate Initialization Card followed by the Data Cards is required for each list of single-subscripted Permanent Attributes. If the initial values are to be set equal to zero, one or more lists of single-subscripted Permanent Attributes may be handled by a single Initialization Card, providing the lists are of the same length and have consecutive Array Numbers.

To read in the initial values of a list of single-subscripted Permanent Attributes, the Initialization Card entries shown in Fig. 6 are required.

INITIALIZATION CARDS

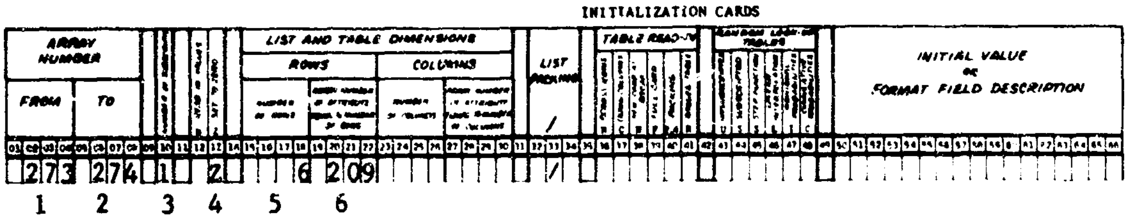


1 2 3 4 5 6

1. Enter the Array Number in Cols. 1 through 4.
2. Enter a "1" in Col. 10.
3. Enter an "R" in Col. 12.
4. In Cols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 5 below.
5. In Cols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is preset for each table. It is preprinted in Table 7.
6. In Cols. 50 through 66, enter a single FORMAT statement Field Description enclosed in parentheses and preceded by an optional constant, if desired. This Field Description tells how the initial values of the list are to appear in the subsequent Data Cards. Each Data Card will be read starting in Col. 1. If desired, successive values may appear across the Data Card.

Fig. 6 -- Initialization Card Entries for Reading In a Single-subscripted Permanent Attribute List

One or more lists of single-subscripted Permanent Attributes describing the same Entity and having consecutive Array Numbers can be initially set equal to zero by the Initialization Card entries shown in Fig. 7. Inserting the letter "Z" in Col. 13 causes zeros to be stored in the entire word.

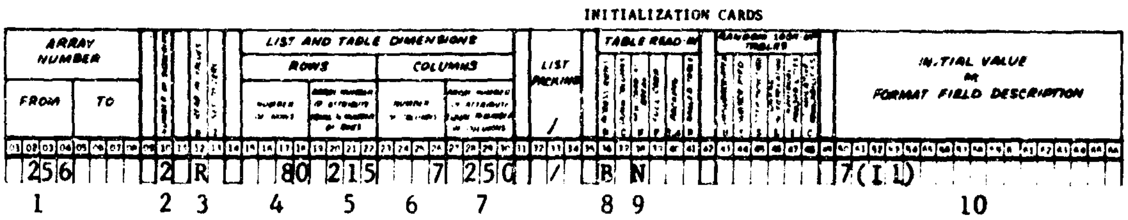


1. Enter the lowest Array Number in Cols. 1 through 4.
2. Enter the highest Array Number in Cols. 5 through 8.
3. Enter a "1" in Col. 10.
4. Enter a "Z" in Col. 13.
5. In Cols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 6 below.
6. In Cols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is present for each table. It is preprinted in Table 7.

Fig. 7 -- Initialization Card Entries for Setting Single-subscripted Permanent Attribute Lists to Zero

Double-subscripted Permanent Attributes. If non-zero initial values are to be read in for a table of double-subscripted Permanent Attributes, each table requires a separate Initialization Card followed by Data Cards containing the values. However, a single Initialization Card may serve to zero out one or more Attribute tables, providing they all describe the same pair of Permanent Entities and have consecutive Array Numbers. The procedure for setting Ragged Tables equal to zero is described below.

Figure 8 shows the Initialization Card entries required for reading in the initial values of a table of double-subscripted Permanent Attributes.



1. Enter the Array Number in Cols. 1 through 4.
2. Enter a "2" in Col. 10.
3. Enter an "R" in Col. 12.
4. In Cols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 5 below.
5. In Cols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is present for each table. It is preprinted in Table 7.
6. In Cols. 23 through 26, indicate the largest column subscript.
7. In Cols. 27 through 30, enter the Array Number of the System Variable, the value of which is equal to the value of the largest <u>column</u> subscript.
8. Indicate the order in which the Attribute values are to be read from the Data Cards by entering an "B" in Col. 36 if the values are to be read across rows, or entering a "C" in Col. 37 if they are to be read down columns.
9. If the beginning of each new row or column is to start on a new Data Card, enter an "N" in Col. 38. If, instead of starting on a new card, the first entry in a new row or column immediately follows the last entry in the preceding row or column, put an "F" in Col. 39.
10. In Cols. 50 through 66, enter a FORMAT statement Field Description enclosed in parentheses indicating how the table entries are to appear in subsequent Data Cards.

Fig. 8 -- Initialization Card Entries for Reading in a Double-subscripted Permanent Attribute Table

<u>Initial Conditions</u>. SIMSCRIPT requires that all permanent system variables be given initial values in ascending order of their Numbers (1-N). Data Deck Card 2 to the end (the last array number "N") is the initial conditions deck.

With each output program is a Variable Description and Initialization Table to specify the initial value(s) assigned to each permanent

system variable. The Formats used to initialize the different types of variables (e.g., unsubscripted, single-subscripted, double-subscripted) have been previously described. There are no exogenous events used in any of the report generators.

Part 2

PROGRAM LIBRARY

Program 1

NOR TIME SUMMARY

I. NOR TIME SUMMARY

The NOR (Not Operationally Ready) time summary (Fig. 9) is a li..'ng of the downtime distributions for any set of fail levels and a count of the demands for the simulated fleet. The program may be initialized to specify any time period desired as well as any portion of the simulated fleet, i.e., a single base or all bases. This output is shown in the following figure along with an explanation of each of the columns of the output listing.

INITIALIZATION

Table 1 lists the Initialization requirements. Only six arrays require inputs. Array 23 is the number of bases to be analyzed. Array 24 lists the base (quantity specified in Array 23) numbers. Array 26 is the quantity of different failure levels to be counted in the NOR time. Array 27 lists the failure level numbers. Array 33 is the time that the reports are to end. Array 53 is the Report Interval, which specifies the time period at which the data are to be accumulated and printed (the example data are initialized for 1-day reports). All of the other arrays are set to zero.

For the initialization formats, the user may use the example data contained with the program listing or refer to Section 2, Initialization instructions for unsubscripted and single-subscripted system variables.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR.)

When a label record is READ from the input tape, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may run at one time.

..........NOR TIME REPORT..........

PERIOD	TIME THIS PER	TIME TO DATE	THIS PER IN	THIS PER OUT	TO DATE IN	TO DATE OUT	THIS PERIOD MIN	THIS PERIOD MAX	THIS PERIOD AVG	THIS PERIOD STD DEV	TO DATE MIN	TO DATE MAX	TO DATE AVG	TO DATE STD DEV
1.00	3.7424	3.7424	5	0	5	0	0.2474	0.9664	0.7486	0.6911	0.0000	0.9664	0.7486	0.6911
2.00	2.1202	5.8631	3	5	8	5	0.0278	1.0000	0.2650	0.4244	0.0280	1.2474	0.7329	0.7168
3.00	1.8443	7.7074	2	3	10	8	0.045	1.0000	0.3689	0.4756	0.0280	2.2474	0.7707	0.7465
4.00	1.2201	8.9276	3	3	13	11	0.036.	1.0000	0.2440	0.3921	0.0280	3.2474	0.6867	0.7162
5.00	1.4878	10.4154	2	2	15	13	0.0356	1.0000	0.3720	0.4443	0.0280	4.2474	0.6944	0.7222
6.00	1.1887	11.6041	2	2	17	15	0.0254	1.0000	0.2972	0.4241	0.0280	5.2474	0.6826	0.7216
7.00	2.5244	14.1285	3	2	20	17	0.0240	1.0000	0.5049	0.5851	0.0240	6.2474	0.7064	0.7431
8.00	3.6161	17.7446	2	1	22	18	0.0758	1.0000	0.7236	0.7178	0.0240	7.2474	0.8067	0.8046
9.00	3.9907	21.7373	4	4	26	22	0.0011	1.0000	0.4988	0.6345	0.0011	8.2474	0.8360	0.8305
10.00	4.4605	26.1978	3	2	29	24	0.0298	1.0000	0.6372	0.7024	0.0011	9.2474	0.9334	0.8701
11.00	3.2644	29.4627	3	5	32	29	0.0357	1.0000	0.4056	0.5709	0.0011	10.2474	0.9231	0.8830
12.00	3.4647	32.9274	1	0	33	29	0.4047	1.0000	0.8712	0.7687	0.0011	11.2474	0.9978	0.9229
13.00	4.4686	37.4262	1	0	34	29	0.4988	1.0000	0.8998	0.8184	0.0011	12.2474	1.1308	0.9749
14.00	4.3392	41.4654	0	1	34	30	0.0392	1.0000	0.8078	0.7992	0.0011	13.2474	1.2196	1.0319
15.00	4.8953	46.3607	1	0	35	30	0.0953	1.0000	0.9791	0.8744	0.0011	14.2474	1.3246	1.0801
16.00	5.8383	52.1995	2	1	37	31	0.0748	1.0000	0.8341	0.8242	0.0011	15.2474	1.4138	1.1191
17.00	5.0005	57.2000	0	1	37	32	0.0005	1.0000	0.8467	0.8326	0.0011	16.2474	1.5481	1.1764
18.00	5.6129	62.8926	1	0	38	32	0.6129	1.0000	0.9355	0.8602	0.0011	17.2474	1.6551	1.2191
19.00	5.0827	67.9755	0	1	38	33	0.0827	1.0000	0.8471	0.8326	0.0011	18.2474	1.7888	1.2706
20.00	5.7763	73.7519	1	0	39	33	0.7763	1.0000	0.9627	0.8794	0.0011	19.2474	1.8911	1.3092
21.00	1.9626	75.7142	11	15	50	48	0.0279	0.6759	0.1154	0.1873	0.0011	19.4032	1.5143	1.1657
22.00	2.5128	78.2270	1	0	51	48	0.5120	1.0000	0.8376	0.6893	0.0011	19.4032	1.5339	1.1731
23.00	4.2385	82.4656	4	2	55	50	0.0109	1.0000	0.6055	0.6693	0.0011	19.4032	1.4994	1.1596
24.00	3.9564	86.4220	1	2	56	52	0.0481	1.0000	0.6594	0.7140	0.0011	19.4032	1.5432	1.1774
25.00	3.0126	89.4345	0	1	56	53	0.0126	1.0000	0.7531	0.7495	0.0011	19.4032	1.5970	1.1995
26.00	5.9504	95.3849	1	0	57	53	0.9504	1.0000	0.9876	0.8538	0.0011	19.4032	1.6383	1.2171
27.00	4.8314	98.2163	1	0	58	53	0.0314	1.0000	0.9663	0.8631	0.0011	19.4032	1.6934	1.2393
28.00	5.4026	103.6189	1	0	59	53	0.4026	1.0000	0.9004	0.8428	0.0011	19.4032	1.7563	1.2634
29.00	5.1181	108.7370	[illegible]	2	60	55	0.0497	1.0000	0.7312	0.7816	0.0011	19.4032	1.8123	1.2859
30.00	5.4285	114.1655	1	0	61	55	0.4285	1.0000	0.9047	0.8445	0.0011	19.4032	1.8716	1.3079
31.00	5.1078	119.2733	1	2	62	57	0.0461	1.0000	0.7297	0.7816	0.0011	19.4032	1.9238	1.3272
32.00	5.0441	124.3173	1	1	63	58	0.0441	1.0000	0.8407	0.8328	0.0011	19.4032	1.9733	1.3461
33.00	5.3783	129.6956	1	0	64	58	0.3783	1.0000	0.8964	0.8413	0.0011	19.4032	2.0265	1.3650
34.00	5.6092	135.3047	1	1	65	59	0.0504	1.0000	0.8099	0.8085	0.0011	19.4032	2.0825	1.3944
35.00	5.3563	140.6610	0	1	65	60	0.0563	1.0000	0.8427	0.8327	0.0011	19.4032	2.1603	1.4115

Time period in days

Total NOR time this period

Cumulative NOR time

Number of maintenance demands during the period

Number of maintenance demands satisfied this period

Cumulative number of maintenance demands

Cumulative number of maintenance demands satisfied

NOR time distribution duri.g the period

Cumulative NOR time distribution

Fig. 9 -- NOR Time Summary

Table 1

INITIALIZATION AND DESCRIPTION: NOR TIME SUMMARY

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 3 Col.: 19-22 (rows)	Array Number of Attribute to Be Entered in Fig. 3 Col.: 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1-22	0			Z										
23	0	I			V						Number of bases to be analyzed.	BASES	E	
24	1	I			V			23			Specify each base code to be analyzed.	BASE		A
25	0			Z								BFLAG		A
26	0	I			V						Number of failure levels to be analyzed	FLVLS	E	
27	1	I			V			26			Specify each failure level code to be analyzed.	FLVL		A
28-32	0			Z										
33	0		F		V						Time to end report	TUEND	E	
34-52	0			Z										
53	0		F		V						Reporting interval [illegible] 1-DAY = 1.00000 7-DAY = 7.00000	VTIME	E	
54	0			Z										
55-56	0			Z										

When a label record is read, the value of the failure level is compared with the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

If ETIME is greater than RTIME, REPORT is called and the output is displayed. If ETIME is greater than TMEND, REPORT is called and the output is displayed. The run is then terminated.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-PR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/Wk
SXDW = S/Wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRSM = ID Addresses
MORE = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-PR).

DTLV1 = Integer variable 1
DTLV2 = Integer variable 2
DTLV3 = Integer variable 3
DTLV4 = Integer variable 4
DTLV5 = Integer variable 5
DTLV6 = Integer variable 6
DTLV7 = Integer variable 7
DTLV8 = Integer variable 8
DTLV9 = Float variable 1
DTLV0 = Float variable 2

Base table.

BASES = Number of base codes to be processed.

BASE = Base codes to be processed.

BFLAG = Controls flow of events as a result of EBAS vs. BASES.

Failure level table.

FLVLS = Number of failure codes to be processed.

FLVL = Failure level codes to be processed.

TMEND = Time initialized to end this run prematurely.

Variables used to output display. (Time in decimal days).

D1 = Reporting time

D2 = NOR time this period

D3 = NOR time to date

D4 = Number of NOR IN this period

D5 = Number of NOR OUT this period

D6 = Number of NOR IN to date

D7 = Number of NOR OUT to date

D8 = Min. NOR time this period

D9 = Max. NOR time this period

D10 = Avg. NOR time this period

D11 = Std. dev. for this period

D12 = Min. NOR time to date

D13 = Max. NOR time to date

D14 = Avg. NOR time to date

D15 = Std. dev. to date

Calculation variables.

TIMEN = NOR time for each period.

SUMP2 = Std. dev. this period.

SUMD2 = Std. dev. to date.

RTIME = Reporting time.

VTIME = Reporting time interval.

LINE = Counter used in report phase.

SETS

Name = NQUE used for NOR events. No subscripts. Ranked N1.

Owner = SIMSCRIPT system.

Member = NOR

N1 = Event time - start of this period.

N3 = System code.

N4 = Failure level code.

N8 = Team ID.

N9 = Request ID.

N10 = Site ID.

N11 = IDSUB.

N12 = Event time - start to date.

Name = PQUE used for preventive maintenance events. No subscripts. Ranked on P1.

Owner = SIMSCRIPT system.

Member = PM

P1 = Event time - start PM.

P2 = System code.

P4 = Request ID.

P5 = Site ID.

Name = HQUE used for overhaul events. No subscripts. Ranked on H1.

Owner = SIMSCRIPT system.

Member = OH

H1 = Event time - start of overhaul.

H2 = System code.

H3 = Team ID.

H4 = Request ID

H5 = Site ID.

```
+      1IDSCR  0   I
+      2IDSUM  0   I
+      3SHFT   0   I
+      4DAYW   0   I
+      5SXDW   0   I
+      6EEAS   0   I
+      7VA     0   I
+      8VB     0   I
+      9VC     0   I
+     10TRSM   0   I
+     11MORE   0   I
+     12ETIME  0   F
+     13DTLV1  0   I
+     14DTLV2  0   I
+     15DTLV3  0   I
+     16DTLV4  0   I
+     17DTLV5  0   I
+     18DTLV6  0   I
+     19DTLV7  0   I
+     20DTLV8  0   I
+     21DTLV9  0   F
+     22DTLV0  0   F
+     23BASES  E   I
+     24BASE   1   I
+     25BFLAG  0   I
+     26FLVLS  F   I
+     27FLVL   1   I
+     28FNQUE  0   I
+     29LNQUE  0   I
+     30FPQUE  0   I
+     31LPQUE  0   I
+     32MSIIE  0   I
+     33TMFND  0   F
+     34D1     0   F
+     35D2     0   F
+     36D3     0   F
+     37D4     0   I
+     38D5     0   I
+     39D6     0   I
+     40D7     0   I
+     41D8     0   F
+     42D9     0   F
+     43D10    0   F
+     44D11    0   F
+     45D12    0   F
+     46D13    0   F
+     47D14    0   F
+     48D15    0   F
+     49TIMEN  0   F
+     50SUMP2  0   F
+     51SUMD2  0   F
+     52RTIME  0   F
+     53VTIME  0   F
```

```
+                                    54LINE  0    I
+                                    55FHQUE 0    I
+                                    56LHQUE 0    I
+                                                               NQUEO   *N1     L
+T NOR  8  8       T N1      1    F
+                  T N3      2    I
+                  T N4      4    I
+                  T N8      5    I
+                  T N9      6    I
+                  T N10     7    I
+                  T N11     8    I
+                  T N12    31    F
+                  T PNQUE32      I
+                  T SNQUE33      I
+                                                               PQUE    *P1     L
+T PM    8         T P1      1    F
+                  T P2      2    I
+                  T P4      3    I
+                  T P5      4    I
+                  T PPQUE   5    I
+                  T SPQUE   6    I
+                                                               HQUEO   *N1     L
+T OH    8         T H1      1    F
+                  T H2      2    I
+                  T H3      3    I
+                  T H4      4    I
+                  T H5      5    I
+                  T PHQUE   6    I
+                  T SHQUE   7    I
```

```
*IBFTC MAIN
                MAIN ROUTINE
C
C
C      .....PLANET - NOR TIME SUMMARY.....
C
C
C.....PURPOSE - TO REPORT NOR TIME.
C
C
C.....INPUT   - TAPE FROM ABC MODEL.
C
C
C.....OUTPUT  - PRINTER (SIMSCRIPT RPG).
C
C
C
      REWIND 9
C
      LET RTIME = RTIME + VTIME
      LET D8    =  99999.99999
      LET D12   =  99999.99999
C
    1 CALL RLBL
      CALL SELECT
C
      IF (IDSUB) EQ (    3), GO TO 3
      IF (BFLAG) NE (    0), GO TO 9999
      IF (IDSUB) EQ ( 110), GO TO 110
      IF (IDSUB) EQ ( 200), GO TO 200
      IF (IDSUB) EQ ( 500), GO TO 500
      IF (IDSUB) EQ ( 600), GO TO 600
      IF (IDSUB) EQ (1900), GO TO 1900
      IF (IDSUB) EQ (2000), GO TO 2000
      IF (IDSUB) EQ (2100), GO TO 2100
      IF (IDSUB) EQ (2150), GO TO 2150
      IF (IDSUB) EQ (2300), GO TO 2300
      IF (IDSUB) EQ (2400), GO TO 2400
      IF (IDSUB) EQ (2500), GO TO 2500
C
      GO TO 9999
C
    3 CALL R3
      CALL EXIT
C
  110 CALL R110
      GO TO 9999
C
  200 CALL R200
      GO TO 9999
C
  500 CALL R500
      GO TO 9999
```

```
C
  600 CALL R600
      GO TO 9999
C
 1900 CALL R1900
      GO TO 9999
C
 2000 CALL R2000
      GO TO 9999
C
 2100 CALL R2100
      GO TO 9999
C
 2150 CALL R2150
      GO TO 9999
C
 2300 CALL R2300
      GO TO 9999
C
 2400 CALL R2400
      GO TO 9999
C
 2500 CALL R2500
      GO TO 9999
C
 9999 IF (MORE) EQ (0). GO TO 1
      CALL RDTL
      GO TO 9999
C
      END
```

```
*IBFTC RLBL
               SUBROUTINE RLBL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....LABEL RECORDS.
C
      LET BFLAG = 0
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
C
      STORE I1 IN IDSOR
      STORE I2 IN IDSUB
      STORE I3 IN SHFT
      STORE I4 IN DAYW
      STORE I5 IN SXOW
      STORE I6 IN EBAS
      STORE I7 IN VA
      STORE I8 IN VB
      STORE I9 IN VC
      STORE I10 IN TRSM
      STORE I11 IN MORE
      STORE I12 IN FTIME
C
      IF (FTIME) GR (TMEND), GO TO 3
C
    1 IF (FTIME) GR (RTIME), GO TO 2
      RETURN
C
    2 CALL REPORT
      GO TO 1
C
    3 LET FTIME = TMEND
      CALL R3
      CALL EXIT
C
      END
```

```
*IBFTC SELECT
              SUBROUTINE SELECT
C
C
C.....PURPOSE - TO SELECT EVENTS BY BASE.
C
C
         DO TO 1, FOR EACH BASES I
         IF (EBAS) EQ (BASE(I)), GO TO 2
     1 LOOP
         LET BFLAG = 1
         GO TO 9999
C
     2 LET BFLAG = 0
         GO TO 9999
C
  9999 RETURN
         END
```

```
*IBFTC RDTL
              SUBROUTINE RDTL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....DETAIL RECORDS.
C
X        READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
C
         STORE I1 IN DTLV1
         STORE I2 IN DTLV2
         STORE I3 IN DTLV3
         STORE I4 IN DTLV4
         STORE I5 IN DTLV5
         STORE I6 IN DTLV6
         STORE I7 IN DTLV7
         STORE I8 IN DTLV8
         STORE I9 IN DTLV9
         STORE I10 IN DTLV0
C
         LET MORE = MORE - 1
C
         RETURN
         END
```

```
$IBFTC R3
              SUBROUTINE R3
C
C
C.....PURPOSE - TO CLOSE-OUT AND END R-PHASE.
C
C
C.....IDSUB = 3.
C
C
      LET LINE = 0
      LET RTIME = ETIME
C
   10 DO TO 3, FOR EACH M IN NQUE
C
      REMOVE M FROM NQUE
C
      LET TIMEN = ETIME - N1(M)
      LET TIMEM = ETIME - N12(M)
C
      LET D5 = D5 + 1
      LET D7 = D7 + 1
      LET D2 = D2 + TIMEN
      LET D3 = D3 + TIMEN
C
      LET SUMP2 = SUMP2 + TIMEN**2
      LET SUMD2 = SUMD2 + TIMEN**2
C
      IF (TIMEN) GE (D8), GO TO 1
      LET D8 = TIMEN
C
    1 IF (TIMEN) LE (D9), GO TO 2
      LET D9 = TIMEN
C
    2 IF (TIMEM) LE (D13), GO TO 3
      LET D13 = TIMEM
C
    3 REPEAT 10
C
      CALL REPORT
C
      REWIND 3
C
      RETURN
      END
```

```
*IBFTC R110
                 SUBROUTINE R110
 C
 C
 C.....PURPOSE - START NOR FOR EXOG. FAILURE, EXOG. PM, EXOG. OVERHAUL.
 C
 C
 C.....IDSUB = 110.
 C
 C
       IF (VC) EQ (2), GO TO 2
       IF (VC) EQ (4), GO TO 4
       IF (VC) EQ (6), GO TO 6
       GO TO 9999
 C
 C..........EXOG. FAILURE.
 C
     2 DO TO 20, FOR EACH FLVLS I
       IF (VB) EQ (FLVL(I)), GO TO 21
    20 LOOP
       GO TO 9999
 C
    21 CREATE NOR CALLED N
       STORE ETIME IN N1(N)
       STORE ETIME IN N12(N)
       STORE VA    IN N3(N)
       STORE VP    IN N4(N)
       STORE TRSM  IN N10(N)
       STORE IDSUB IN N11(N)
       FILE N IN NOUE
       LET D4 = D4 + 1
       LET D6 = D6 + 1
       GO TO 9999
 C
 C..........EXOG. PM.
 C
     4 CREATE PM CALLED P
       STORE ETIME IN P1(P)
       STORE VA    IN P2(P)
       STORE TRSM  IN P5(P)
       FILE P IN POUE
       GO TO 9999
 C
 C..........EXOG. OVERHAUL.
 C
     6 CREATE OH CALLED H
       STORE ETIME IN H1(H)
       STORE VA    IN H2(H)
       STORE TRSM  IN H5(H)
       FILE H IN HOUE
       GO TO 9999
 C
  9999 RETURN
       END
```

```
*IBFTC R200
              SUBROUTINE R200
C
C
C.....PURPOSE - TO END NOR FOR MAINTENANCE COMPLETED.
C
C
C.....IDSUB = 200.
C
C
       IF (MORE) EQ (0), GO TO 9999
       CALL RDTL
C
       FIND FIRST, FOR EACH M IN NQUE, WITH (N8(M)) EQ (TRSM),
      X                                    AND (N10(M)) EQ (VA),
      XWHERE IN, IF NONE, GO TO 9999
C
       LET TIMEN = FTIME - N1(IN)
       LET TIMEM = FTIME - N12(IN)
C
       LET D5 = D5 + 1
       LET D7 = D7 + 1
       LET D2 = D2 + TIMEN
       LET D3 = D3 + TIMEN
C
       LET SUMP2 = SUMP2 + TIMEN**2
       LET SUMD2 = SUMD2 + TIMEN**2
C
       IF (TIMEN) GE (D8), GO TO 1
       LET D8 = TIMEN
C
     1 IF (TIMEN) LE (D9), GO TO 2
       LET D9 = TIMEN
C
     2 IF (TIMEM) GE (D12), GO TO 3
       LET D12 = TIMEM
C
     3 IF (TIMEM) LE (D13), GO TO 4
       LET D13 = TIMEM
C
     4 REMOVE IN FROM NQUE
       DESTROY NOR CALLED IN
C
  9999 RETURN
       END
```

```
*IBFTC R500
              SUBROUTINE R500
C
C
C.....PURPOSE - TEAM DISPATCH BY BASE CONTROL.
C
C
C.....IDSUP = 500.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      IF (DTLV4) EQ (MSITE), GO TO 9999
C
      FIND FIRST, FOR EACH M IN NODE, WITH (N9(M)) EQ (VC),
     X                                 AND (N10(M)) EQ (DTLV1),
     XWHERE IN, IF NONE, GO TO 1
C
      STORE TRSM IN N8(IN)
      GO TO 9999
C
    1 FIND FIRST, FOR EACH M IN HODE, WITH (H4(M)) EQ (VC),
     X                                 AND (H5(M)) EQ (DTLV1),
     XWHERE IH, IF NONE, GO TO 9999
C
      STORE TRSM IN H3(IH)
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IPETC R600
              SUBROUTINE R600
C
C
C.....PURPOSE - TEAM ARRIVAL AT SITE.
C
C
C.....IDSUB = 600.
C
C
      IF (VB) NE (6), GO TO 9999
C
      FIND FIRST, FOR EACH M IN HQUE, WITH (H5(M)) EQ (TRSM),
     X                                  AND (H5(M)) EQ (VA),
     XWHERE IH, IF NONE, GO TO 9999
C
      CREATE NQR CALLED N
      STORE FTIME  IN N1(N)
      STORE FTIME  IN N12(N)
      STORE F2(IH) IN N3(N)
      STORE H3(IH) IN N8(N)
      STORE H4(IH) IN N9(N)
      STORE H5(IH) IN N10(N)
      STORE IDSUB  IN N11(N)
C
      FILE N IN NQUE
C
      LET D4 = D4 + 1
      LET D6 = D6 + 1
C
      REMOVE IH FROM HQUE
      DESTROY OH CALLED IH
C
 9999 RETURN
      END


*IEFTC R1900
              SUBROUTINE R1900
C
C
C.....PURPOSE - GENERATE PROPER MSITE FOR RETURNING TEAMS. (R5[illegible])
C
C
C.....IDSUB = 1900.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      IF (VC) GE (MSITE), LET MSITE = VC + 1
C
 9999 RETURN
      END
```

```
*IBFTC R2000
               SUBROUTINE R2000
C
C
C.....PURPOSE - REQUEST FOR PM.
C
C
C.....IDSUB = 2000.
C
C
      CREATE PM CALLED P
C
      STORE ETIME IN P1(P)
      STORE VB    IN P2(P)
      STORE TRSM  IN P5(P)
C
      FILE P IN PQUE
C
 9999 RETURN
      END


*IBFTC R2100
               SUBROUTINE R2100
C
C
C.....PURPOSE - TO START NOR FOR PM.
C
C
C.....IDSUB = 2100.
C
C
      FIND FIRST, FOR EACH M IN PQUE, WITH (P5(M)) EQ (TRSM),
     XAND (P2(M)) EQ (VB), WHERE IP, IF NONE, GO TO 9999
C
      CREATE NOR CALLED N
C
      STORE ETIME  IN N1(N)
      STORE ETIME IN N12(N)
      STORE P2(IP) IN N3(N)
      STORE VC     IN N8(N)
      STORE P4(IP) IN N9(N)
      STORE P5(IP) IN N10(N)
      STORE IDSUB  IN N11(N)
C
      FILE N IN NQUE
C
      LET D4 = D4 + 1
      LET D6 = D6 + 1
C
      REMOVE IP FROM PQUE
      DESTROY PM CALLED IP
C
 9999 RETURN
      END
```

```
*IBFTC R2150
                SUBROUTINE R2150
C
C
C.....PURPOSE - TO START NOR FOR FAILURE CAUSED BY PM.
C
C
C.....IDSUB = 2150.
C
C
          DO TO 1, FOR EACH FLVL, I
          IF (VA) EQ (FLVL(I)), GO TO 2
        1 LOOP
          GO TO 9999
C
        2 CREATE NOR CALLED N
C
          STORE ETIME  IN N1(N)
          STORE ETIME  IN N12(N)
          STORE VB     IN N3(N)
          STORE VA     IN N4(N)
          STORE TRSM   IN N10(N)
          STORE IDSUB  IN N11(N)
C
          FILE N IN NQUE
C
          LET D4 = D4 + 1
          LET D6 = D6 + 1
C
     9999 RETURN
          END
```

```
*IBFTC R2300
              SUBROUTINE R2300
C
C
C.....PURPOSE - TO START NOR FOR ALERT-CONTINUOUS MONITOR.
C
C
C.....IDSUB = 2300.
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 CREATE NOR CALLED N
C
      STORE ETIME IN N1(N)
      STORE ETIME IN N12(N)
      STORE VH    IN N3(N)
      STORE VA    IN N4(N)
      STORE TRSM  IN N10(N)
      STORE IDSUB IN N11(N)
C
      FILE N IN NQUE
C
      LET D4 = D4 + 1
      LET D6 = D6 + 1
C
 9999 RETURN
      END
```

```
*IBFTC R2400
              SUBROUTINE R2400
C
C
C.....PURPOSE - RESOURCE REQUEST.
C
C
C.....IDSUB = 2400.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      IF (DTLVI) EQ (1), GO TO 1
      IF (DTLVI) EQ (2), GO TO 2
      IF (DTLVI) EQ (3), GO TO 3
      IF (DTLVI) EQ (4), GO TO 4
      IF (DTLVI) EQ (5), GO TO 5
      IF (DTLVI) EQ (6), GO TO 6
      GO TO 9999
C
    1 FIND FIRST, FOR EACH M IN NCUE, WITH (N11(M)) EQ (2300),
     XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
     XGO TO 9999
C
      STORE VC IN N9(IN)
      GO TO 9999
C
    2 FIND FIRST, FOR EACH M IN NCUE, WITH (N11(M)) EQ (110),
     XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
     XGO TO 9999
C
      STORE VC IN N9(IN)
      GO TO 9999
C
    3 FIND FIRST, FOR EACH M IN NCUE, WITH (N11(M)) EQ (2500),
     XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
     XGO TO 9999
C
      STORE VC IN N9(IN)
      GO TO 9999
C
    4 FIND FIRST, FOR EACH M IN PCUE, WITH (P5(M)) EQ (TRSM),
     XAND (P2(M)) EQ (VB), WHERE IP, IF NONE, GO TO 9999
C
      STORE VC IN P4(IP)
      GO TO 9999
C
    5 FIND FIRST, FOR EACH M IN NCUE, WITH (N11(M)) EQ (2150),
     XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
     XGO TO 9999
C
      STORE VC IN N9(IN)
```

```
      GO TO 9999
C
    6 FIND FIRST, FOR EACH M IN HQUE, WITH (H5(M)) EQ (TRSM),
     XAND (H2(M)) EQ (VB), WHERE IH, IF NONE, GO TO 9999
C
      STORE VC IN H4(IH)
      GO TO 9999
C
 9999 RETURN
      END




*IBFTC R2500
              SUBROUTINE R2500
C
C
C.....PURPOSE - TO START NOR FOR UNDETERMINED FAILURE.
C
C
C.....IDSUB = 2500.
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 CREATE NOR CALLED N
C
      STORE ETIME IN N1(N)
      STORE ETIME IN N12(N)
      STORE VB    IN N3(N)
      STORE VA    IN N4(N)
      STORE TRSM  IN N10(N)
      STORE IDSUB IN N11(N)
C
      FILE N IN NQUE
C
      LET D4 = D4 + 1
      LET D6 = D6 + 1
C
 9999 RETURN
      END
```

```
*IBFTC REPORT
                SUBROUTINE REPORT
C
C
C.....PURPOSE - TO REPORT NOR TIME AS OF A GIVEN TIME.
C
C
C.....CALLED BY RLBL OR R3.
C
C
      LET IP = D5
      LET ID = D7
C
  200 DO TO 299, FOR EACH M IN NQUE
C
      LET TIMEN = RTIME - N1(M)
      LET TIMEM = RTIME - N12(M)
      LET N1(M) = RTIME
C
      LET D2 = D2 + TIMEN
      LET D3 = D3 + TIMEN
C
      LET SUMP2 = SUMP2 + TIMEN**2
      LET SUMD2 = SUMD2 + TIMEN**2
C
      LET IP = IP + 1
      LET ID = ID + 1
C
      IF (TIMEN) GE (D8), GO TO 201
      LET D8 = TIMEN
C
  201 IF (TIMEN) LE (D9), GO TO 202
      LET D9 = TIMEN
C
  202 IF (TIMEM) LE (D13), GO TO 299
      LET D13 = TIMEM
C
  299 REPEAT 200
C
      IF (LINE) EQ (0), GO TO 2
      IF (LINE) EQ (50), GO TO 2
C
    1 LET D1 = RTIME
      IF ((SUMP2 - (FLOAT(IP) * (D2 / FLOAT(IP)**2))) / (FLOAT(IP)))
     X10,10,11
   10 LET D11 = 0.0
      GO TO 12
   11 LET D11 = SQRT((SUMP2 - (FLOAT(IP) * (D2 / FLOAT(IP)**2)))
     X         / (FLOAT(IP)))
   12 IF ((SUMD2 - (FLOAT(ID) * (D3 / FLOAT(ID)**2))) / (FLOAT(ID)))
     X13,13,14
   13 LET D15 = 0.0
      GO TO 15
```

```
   14 LET D15 = SQRT((SUMD2 - (FLOAT(ID) * (D3 / FLOAT(ID)**2)))
     X          / (FLOAT(ID)))
   15 LET D10 = D2 / FLOAT(IF)
      LET D14 = D3 / FLOAT(ID)
C
      IF (D8) GR (99999.0), LET D8 = 0.0
      LET S12 = D12
      IF (D12) GR (99999.0), LET D12 = 0.0
C
      CALL DTAIL
C
      LET LINE = LINE + 1
C
      LET PTIME = RTIME + VTIME + .00001
C
      LET D2 = 0.0
      LET D4 = 0
      LET D5 = 0
      LET D8 = 99999.99999
      LET D9 = 0.0
      LET D12 = S12
      LET SUMP2 = 0.0
C
      GO TO 9999
C
    2 CALL HDING
      LET LINE = 0
      GO TO 1
C
 9999 RETURN
      END


*IBFTC HDING
      REPORT HDING
*     ..........NOR TIME REPORT..........
*                    TIME          TIME         THIS PER    TO DATE .........T H I
*        PERIOD  THIS PER      TO DATE       IN   OUT    IN  OUT      MIN          M
      END
                                                                                    *
                                                                               2
 S   P E R I O D........... ..............T O   D A T E..............
AX        AVG       STD DEV       MIN         MAX          AVG       STD DEV    1
      END



*IBFTC DTAIL
      REPORT DTAIL
*        ****.** ****.****  ****.****  **** **** **** **** ****.**** ****
 *    D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15
      END
                                                                                    *
.**** ****.**** ****.**** ****.**** ****.**** ****.**** ****.****    *
                                                                                    *
      END
```

INITIALIZATION DECK

```
*ENTRY          MAIN
1           56                                              99
   1  22 0  7
  23      0 R                                                    100
  24      1 R   100  23                                          (12)
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
25 0 1

```
   26      0 R                                               10
   27      1 R   10  26                                     (11)
0
1
2
3
4
5
6
7
8
9
  28   32 0   Z
  33      0 R
  34   52 0   Z                                    999.99999
  53      0 R
  54      0   Z                                      0.99999
  55   56 0   Z
```

Program 2

WEAPON SYSTEM AVAILABILITY

II. WEAPON SYSTEM AVAILABILITY

The Weapon System Availability program can be used with either aircraft or missiles. It displays the off-alert time by tail number as well as a chronological, time-oriented listing of what happened while the site was off alert. This output is shown in Fig. 10. An explanation of each of the column headings follows.

Event Time: The simulated time at which the various events take place.

Tail No.: The tail number of the site.

Site Type: The site type number specified in the ABC simulator Exogenous Event Genera data card, Col. 13.

SQUAD: Squadron or Base number that was specified in the ABC simulator Exogenous Event Genera data card, Cols. 19-20.

Init Stats: Indicates the initial status that removed the site from the available status. Codes are as follows:

AA = Detected failure
AB = Latent failure
ZZ = Exogenous failure
OH = Unit removed for overhaul or time replacement
01-12 = Periodic Maintenance number

Total Off-Alert: Indicates the actual amount of time (days, hours, minutes) that the site is (not) available.

Evnt Stat: Indicates the events that occurred while the site was not available. Codes are as follows:

AA = Detected failure
AB = Latent failure
ZZ = Exogenous failure
OH = Unit removed for overhaul or time replacement
01-12 = Periodic Maintenance number
AD = Demand
AT = Team Dispatched by Base Control
AS = Team Arrives at Site
XA = Maintenance Successfully completed.

FD = Personnel failure during maintenance

FE = Equipment failure during maintenance

FS = Spare part failure during maintenance

SYST FAIL: Indicates the Unit responsible for the demand.

FAIL LEVL: Failure level.

LAG TIME: The time the maintenance team arrives at the site minus the time the team is dispatched (AS - AT) in days, hours, minutes.

DUR TIME: The time that the site is not available. The time that the maintenance is successfully completed minus the time that the site went off alert. (XA-AA) or (XA-AB) or (XA-ZZ) or (XA-OH) or (XA-01 to 12).

ALERT Deg. Time: The time that maintenance was successfully completed minus the time of the failure. In the event of multiple failures, the subsequent degredation time will be from fix time to fix time.

SEQ No: Self-explanatory.

TID: Team Identification number used only for program checkout.

RID: Request Identification number used for program checkout.

SID: Site Identification number used for program checkout.

EVNT: Label Record number printed by the simulation phase.

The last page of the report will display the status of the sites that remain not available at the end of simulation.

EVENT TIME	TAIL NO	SITE TYPE	SQUAD	INIT STAT	TOTAL OFF-ALERT	EVNT STAT	SYST FAIL	FAIL LEVL	LAG TIME	DUR. TIME	ALERT DEG. TIME	SEQ NO.	TID	RID	SID	EVNT
10 0 0	9	1	2		0 0 0	AT	1	6	0 0 0	0 0 0	0 0 0	4	0	0	0	500
10 0 2	9	1	2		0 0 0	AS	1	6	0 0 0	0 0 0	0 0 0	5	0	0	0	600
10 1 40	9	1	2		0 0 0	XA	1	6	0 0 0	0 0 0	0 0 0	6	0	0	0	200
10 12 0	10	1	2	ZZ	0 0 52	ZZ	2	2	0 0 2	0 0 52	0 0 52	1	0	0	0	110
10 12 0	10	1	2		0 0 0	AD	2	2	0 0 0	0 0 0	0 0 0	2	0	0	0	100
10 12 0	10	1	2		0 0 0	AT	2	2	0 0 0	0 0 0	0 0 0	3	0	0	0	500
10 12 2	10	1	2		0 0 0	AS	2	2	0 0 0	0 0 0	0 0 0	4	0	0	0	600
10 12 52	10	1	2		0 0 0	XA	2	2	0 0 0	0 0 0	0 0 0	5	0	0	0	200
10 12 0	12	2	1	ZZ	0 1 4	ZZ	5	1	0 0 2	0 1 4	0 1 4	1	0	0	0	110
10 12 0	12	2	1		0 0 0	AD	5	1	0 0 0	0 0 0	0 0 0	2	0	0	0	100
10 12 0	12	2	1		0 0 0	AT	5	1	0 0 0	0 0 0	0 0 0	3	0	0	0	500
10 12 2	12	2	1		0 0 0	AS	5	1	0 0 0	0 0 0	0 0 0	4	0	0	0	600
10 13 4	12	2	1		0 0 0	XA	5	1	0 0 0	0 0 0	0 0 0	5	0	0	0	200
10 12 0	11	2	1	ZZ	0 1 8	ZZ	5	1	0 0 1	0 1 8	0 1 8	1	0	0	0	110
10 12 0	11	2	1		0 0 0	AD	5	1	0 0 0	0 0 0	0 0 0	2	0	0	0	100
10 12 0	11	2	1		0 0 0	AT	5	1	0 0 0	0 0 0	0 0 0	3	0	0	0	500
10 12 1	11	2	1		0 0 0	AS	5	1	0 0 0	0 0 0	0 0 0	4	0	0	0	600
10 13 8	11	2	1		0 0 0	XA	5	1	0 0 0	0 0 0	0 0 0	5	0	0	0	200
12 12 2	1	1	1	AB	0 12 55	AB	2	2	0 0 3	0 12 55	0 12 55	1	0	0	0	2500
13 0 0	1	1	1		0 0 0	04	2	2	0 0 0	0 0 0	0 0 0	2	0	0	0	2400
13 0 0	1	1	1		0 0 0	AD	2	2	0 0 0	0 0 0	0 0 0	3	0	0	0	100
13 0 0	1	1	1		0 0 0	AT	2	2	0 0 0	0 0 0	0 0 0	4	0	0	0	500
13 0 3	1	1	1		0 0 0	AS	2	2	0 0 0	0 0 0	0 0 0	5	0	0	0	600
13 0 57	1	1	1		0 0 0	XA	2	2	0 0 0	0 0 0	0 0 0	6	0	0	0	200
15 8 2	13	2	1	OH	0 1 48	OH	5	0	0 0 0	0 1 48	0 1 48	1	0	0	0	110
15 9 50	13	2	1		0 0 0	XA	5	0	0 0 0	0 0 0	0 0 0	2	0	0	0	200
15 5 40	1	1	1	AB	0 20 17	AB	1	6	0 0 1	0 20 17	0 20 17	1	0	0	0	2500
16 0 0	1	1	1		0 0 0	04	1	6	0 0 0	0 0 0	0 0 0	2	0	0	0	2400
16 0 0	1	1	1		0 0 0	AD	1	6	0 0 0	0 0 0	0 0 0	3	0	0	0	100
16 0 0	1	1	1		0 0 0	AT	1	6	0 0 0	0 0 0	0 0 0	4	0	0	0	500
16 0 1	1	1	1		0 0 0	AS	1	6	0 0 0	0 0 0	0 0 0	5	0	0	0	609
16 1 56	1	1	1		0 0 0	XA	1	6	0 0 0	0 0 0	0 0 0	6	0	0	0	200
17 9 18	9	1	2	AB	0 16 42	AB	1	6	0 0 2	0 16 42	0 16 42	1	0	0	0	2500
18 0 0	9	1	2		0 0 0	04	1	6	0 0 0	0 0 0	0 0 0	2	0	0	0	2400
18 0 0	9	1	2		0 0 0	AD	1	6	0 0 0	0 0 0	0 0 0	3	0	0	0	109
18 0 0	9	1	2		0 0 0	AT	1	6	0 0 0	0 0 0	0 0 0	4	0	0	0	500
18 0 2	9	1	2		0 0 0	AS	1	6	0 0 0	0 0 0	0 0 0	5	0	0	0	600
18 2 0	9	1	2		0 0 0	XA	1	6	0 0 0	0 0 0	0 0 0	6	0	0	0	200
20 0 1	3	1	1	01	0 0 44	01	1	0	0 0 0	0 0 44	0 0 44	1	0	0	0	2100
20 0 44	3	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	2	0	0	0	200
20 0 1	18	2	2	01	0 1 21	01	1	0	0 0 0	0 0 46	0 0 46	1	0	0	0	2100
20 0 1	18	2	2		0 0 0	01	5	2	0 0 1	0 1 21	0 0 35	2	0	0	0	2150
20 0 1	18	2	2		0 0 0	AD	5	2	0 0 0	0 0 0	0 0 0	3	0	0	0	100
20 0 1	18	2	2		0 0 0	AT	5	2	0 0 0	0 0 0	0 0 0	4	0	0	0	500
20 0 1	18	2	2		0 0 0	AS	5	2	0 0 0	0 0 0	0 0 0	5	0	0	0	600
20 0 46	18	2	2		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	6	0	0	0	200

Fig. 10 -- Weapon System Availability

EVENT TIME	TAIL NO	SITE TYPE	SQUAD	INIT STAT	TOTAL OFF-ALERT	EVNT STAT	SYST FAIL	FAIL LEVL	LAG TIME	DUR. TIME	ALERT DEG. TIME	SEQ NO.	TID	RID	SID	EVNT
20 1 21	18	2	2		0 0 0	XA	5	2	0 0 0	0 0 0	0 0 0	7	0	0	0	200
6 12 53	6	1	1	AB	13 12 29	AB	2	2	0 0 2	13 11 53	13 11 53	1	0	0	0	2500
20 0 2	6	1	1		0 0 0	01	1	0	0 0 0	0 1 21	0 0 37	2	0	0	0	2100
20 0 2	6	1	1		0 0 0	01	2	2	0 0 0	0 0 0	0 0 0	3	0	0	0	2400
20 0 2	6	1	1		0 0 0	AD	2	2	0 0 0	0 0 0	0 0 0	4	0	0	0	100
20 0 2	6	1	1		0 0 0	AT	2	2	0 0 0	0 0 0	0 0 0	5	0	0	0	500
20 0 3	6	1	1		0 0 0	AS	2	2	0 0 0	0 0 0	0 0 0	6	0	0	0	600
20 0 46	6	1	1		0 0 0	XA	2	2	0 0 0	0 0 0	0 0 0	7	0	0	0	200
20 1 22	6	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	8	0	0	0	200
5 23 24	5	1	1	AB	14 2 0	AB	4	4	0 0 1	14 1 39	14 1 39	1	0	0	0	2500
20 0 1	5	1	1		0 0 0	01	1	0	0 0 0	0 1 23	0 0 21	2	0	0	0	2100
20 0 1	5	1	1		0 0 0	01	4	4	0 0 0	0 0 0	0 0 0	3	0	0	0	2400
20 0 1	5	1	1		0 0 0	AD	4	4	0 0 0	0 0 0	0 0 0	4	0	0	0	100
20 0 46	5	1	1		0 0 0	AT	4	4	0 0 0	0 0 0	0 0 0	5	0	0	0	500
20 0 46	5	1	1		0 0 0	AS	4	4	0 0 0	0 0 0	0 0 0	6	0	0	0	600
20 1 2	5	1	1		0 0 0	XA	4	4	0 0 0	0 0 0	0 0 0	7	0	0	0	200
20 1 23	5	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	8	0	0	0	200
19 5 23	9	1	2	AB	0 20 3	AB	1	6	0 0 2	0 20 3	0 20 3	1	0	0	0	2500
20 0 0	9	1	2		0 0 0	04	1	6	0 0 0	0 0 0	0 0 0	2	0	0	0	2400
20 0 0	9	1	2		0 0 0	AD	1	6	0 0 0	0 0 0	0 0 0	3	0	0	0	100
20 0 0	9	1	2		0 0 0	AT	1	6	0 0 0	0 0 0	0 0 0	4	0	0	0	500
20 0 2	9	1	2		0 0 0	AS	1	6	0 0 0	0 0 0	0 0 0	5	0	0	0	600
20 1 25	9	1	2		0 0 0	XA	1	6	0 0 0	0 0 0	0 0 0	6	0	0	0	200
11 12 23	8	1	2	AB	8 13 17	AB	4	4	0 0 2	8 12 55	8 12 55	1	0	0	0	2500
14 2 31	8	1	2		0 0 0	AB	1	6	0 0 1	5 22 10	0 0 0	2	0	0	0	2500
20 0 2	8	1	2		0 0 0	01	1	0	0 0 0	0 1 37	0 0 22	3	0	0	0	2100
20 0 2	8	1	2		0 0 0	01	1	6	0 0 0	0 0 0	0 0 0	4	0	0	0	2400
20 0 2	8	1	2		0 0 0	01	4	4	0 0 0	0 0 0	0 0 0	5	0	0	0	2400
20 0 2	8	1	2		0 0 0	AD	1	6	0 0 0	0 0 0	0 0 0	6	0	0	0	100
20 0 2	8	1	2		0 0 0	AT	1	6	0 0 0	0 0 0	0 0 0	7	0	0	0	500
20 0 2	8	1	2		0 0 0	AD	4	4	0 0 0	0 0 0	0 0 0	[illegible]	0	0	0	100
20 0 2	8	1	2		0 0 0	AT	4	4	0 0 0	0 0 0	0 0 0	9	0	0	0	500
20 0 2	8	1	2		0 0 0	AS	1	6	0 0 0	0 0 0	0 0 0	10	0	0	0	600
20 0 4	8	1	2		0 0 0	AS	4	4	0 0 0	0 0 0	0 0 0	11	0	0	0	600
20 0 41	8	1	2		0 0 0	XA	1	6	0 0 0	0 0 0	0 0 0	12	0	0	0	200
20 1 17	8	1	2		0 0 0	XA	4	4	0 0 0	0 0 0	0 0 0	13	0	0	0	200
20 1 39	8	1	2		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	14	0	0	0	200
20 0 1	2	1	1	01	0 1 42	01	1	0	0 0 0	0 1 42	0 1 42	1	0	0	0	2100
20 1 42	2	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	2	0	0	0	200
20 0 1	1	1	1	01	0 1 56	01	1	0	0 0 0	0 1 56	0 1 56	1	0	0	0	2100
20 1 57	1	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	2	0	0	0	200
8 18 4	4	1	1	AB	19 9 41	AB	3	4	0 0 4	19 9 41	19 9 41	1	0	0	0	2500
20 0 2	4	1	1		0 0 0	01	1	0	0 0 0	0 0 43	0 0 0	2	0	0	0	2100
20 0 2	4	1	1		0 0 0	01	3	4	0 0 0	0 0 0	0 0 0	3	0	0	0	2400
20 0 2	4	1	1		0 0 0	AD	3	4	0 0 0	0 0 0	0 0 0	4	0	0	0	100
20 0 2	4	1	1		0 0 0	AT	3	4	0 0 0	0 0 0	0 0 0	5	0	0	0	500
20 0 2	4	1	1		0 0 0	AS	3	4	0 0 0	0 0 0	0 0 0	6	0	0	0	600
20 0 45	4	1	1		0 0 0	XA	1	0	0 0 0	0 0 0	0 0 0	7	0	0	0	200

Fig. 10 -- Continued

EVENT TIME	TAIL NO	SITE TYPE	SQUAD	INIT STAT	TOTAL OFF-ALERT	EVNT STAT	SYST FAIL	FAIL LEVL	LAG TIME	DUR. TIME	ALERT DEG. TIME	SEQ NO.	TID	RID	SID	EVNT
20 7 47	4	1	1		******** 0 0	00	3	4	0 0 0	0 0 0	0 0 0	1	0	0	29495	2500
25 1 12	5	1	1		******** 0 0	00	4	4	0 0 0	0 0 0	0 0 0	1	0	0	29447	2500
20 13 28	8	1	2		0 22 14	00	4	4	0 0 0	0 0 0	0 0 0	1	0	0	29295	2500
21 11 42	8	1	2		0 [illegible] 0	00	1	6	0 0 0	0 0 0	0 0 0	2	0	0	29295	2500
26 4 3	10	2	2		******** 0 0	00	5	1	0 0 0	0 0 0	0 0 0	1	0	0	29271	2500

Fig. 10 -- Continued

INITIALIZATION

Table 2 lists the Initialization requirements. Only seven Arrays require initialization. Array 23 is the total number of bases to be examined. Array 24 lists the base numbers (quantity specified in Array 25). Array 26 is the quantity of different failure levels to be counted in the report. Array 27 lists the failure level numbers. Array 28 is the quantity of status codes to be included in the report. Array 29 lists the status codes. Array 33 specifies the time for the end of the report. All of the other arrays are set to zero.

The initialization instructions for unsubscripted and subscripted system variables are contained in Part 1, Sec. II of this Memorandum.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes llowed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may be run at one time.

When a label record is read, the value of the failure level is compared to the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

The output display is given whenever a SITE is returned to on-alert status.

If ETIME is greater than TMEND, REPORT is called and the output is displayed. The run is then terminated.

A list of status codes for the STATS table is given on page 44.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Table 2

VARIABLE DESCRIPTION AND INITIALIZATION:
WEAPON SYSTEM AVAILABILITY

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col. 19-22 (rows)	Array Number of Attribute to Be Entered in Fig. 5 Col. 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1-22	0			Z										
23	0	I			V						Number of bases to be analyzed.	BASES	E	
24	1	I			V			23			Specify each base number to be analyzed.	BASE		A
25	0			Z										
26	0	I			V						Number of failure levels to be analyzed.	FLVLS	E	
27	1	I			V			26			Specify each failure level code to be analyzed.	FLVL		A
28	0	I			V						Number of status codes to be analyzed.	STATS	E	
29	1	I			V			28			Specify each status code to be analyzed.	STAT		A
30-32	0			Z										
33	0		F		V						Time to end report.	THEND	E	

Label records (see page 108 of RM-4659-PR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/wk
SXDW = S/wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRSM = ID Addresses
MORE = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-PR).

DTLV1 = Integer variable 1
DTLV2 = Integer variable 2
DTLV3 = Integer variable 3
DTLV4 = Integer variable 4
DTLV5 = Integer variable 5
DTLV6 = Integer variable 6
DTLV7 = Integer variable 7
DTLV8 = Integer variable 8
DTLV9 = Float variable 1
DTLV0 = Float variable 2

Base table.

BASES = Number of base codes to be processed.
BASE = Base codes to be processed.
BFLAG = Controls flow of events as a result of EBAS vs BASES.

Failure level table.

FLVLS = Number of failure codes to be processed.
FLVL = Failure level codes to be processed.

Status code table (See Fig. 10).

STATS = Nu ber of status codes in table.

STAT = Status codes (Alpha).

TMEND = Time initialized to end this run prematurely.

SETS

Name = SQUE used for site events. No subscripts. Ranked on SERNO.

Owner = SIMSCRIPT system.

Member = SITES

SID = Site ID number.

SMODE = Mode of site.

SERNO = Tail number of site.

EBASE = Base number of site.

SINO = Number (1-n) assigned to site by the program.

NFAIL = Number of failures occurring at the site.

Name = EQUE used for site events. One subscript. Ranked on E1.

Owner = SITES.

Member = EVENT (Used for reporting display). (Time in hours, days, minutes).

E1 = Event time.

E2 = Event status.

E3 = System failed.

E4 = Failure level.

E5 = Lag time (Traveling time).

E6 = Duration time (Failure time).

E7 = Alert degradation time.

E8 = Team ID.

E9 = Request ID.

E10 = Site ID.

E11 = IDSUB.

E12 = Initial status.

E13 = Total off-alert time.

E14 = Sequence number of event.

Name = PQUE used for preventive maintenance events. One subscript. Ranked on P1.

Owner = SITES.

Member = PM

P1 = Preventive maintenance request time.

P2 = System requesting PM.

P3 = Regular or exogenous PM flag.

P4 = Request ID.

P5 = Site ID.

P6 = Status code.

Name = HQUE used for overhaul events. One subscript. Ranked on H1.

Owner = SITES.

Member = OH

H1 = Overhaul request time.

H2 = System requesting overhaul.

H3 = Team ID.

H4 = Request ID.

H5 = Site ID.

H6 = Status code.

Status code

STAT = 1-100 are alpha for blank/00, 01-99.

101 is alpha for AA code for a continuous monitor failure.

102 is alpha for AB code for an undetermined failure.

103 is alpha for AD code for a demand at base control.

104 is alpha for AT code for a team dispatch by base.

105 is alpha for AS code for a team arrival at site.

106 is alpha for LO code for a team lost en route.

107 is ha for FE code for an equipment failure at site.

108 is alpha for XA code for maintenance completed.

109 is alpha for ZZ code for exogenous failure.

110 is alpha for OH code for exogenous overhaul.

111 is alpha for FP code for a personnel failure at site.

112 is alpha for FS code for a parts failure at site.

113-200 are not in use.

```
*                                             1IDSUB  0    I
*                                             2IDSUB  0    I
*                                             3SHFT   0    I
*                                             4LAYW   0    I
*                                             5SXDW   0    I
*                                             6FBAS   0    I
*                                             7VA     0    I
*                                             8VB     0    I
*                                             9VC     0    I
*                                            10TRSM   0    I
*                                            11MORE   0    I
*                                            12ETIME  0    F
*                                            13DTLV1  0    I
*                                            14DTLV2  0    I
*                                            15DTLV3  0    I
*                                            16DTLV4  0    I
*                                            17DTLV5  0    I
*                                            18DTLV6  0    I
*                                            19DTLV7  0    I
*                                            20DTLV8  0    I
*                                            21DTLV9  0    F
*                                            22DTLV0  0    F
*                                            23BASES  F    I
*                                            24BASE   1    I
*                                            25BFLAG  0    I
*                                            26FLVLS  E    I
*                                            27FLVL   1    I
*                                            28STATS  F    I
*                                            29STAT   1    I
*                                            30FSQUE  0    I
*                                            31LSQUE  0    I
*                                            32MSITE  0    I
*                                            33TMEND  0    F
*                                                                   SQUEU   *SERNO L
*T SITESR   H          T SID      1     I
*                      T SMODE    2     I
*                      T SERNO    4     I
*                      T FBASE    5     I
*                      T SINO     6     I
*                      T NFAIL    7     I
*                      T PSQUE    8     I
*                      T SSQUE 31       I
*                      T FEQUE 32       I
*                      T LEQUE 33       I
*                      T FPQUE 34       I
*                      T LPQUE 35       I
*                      T FMQUE 36       I
*                      T LMQUE 37       I
*
*                                                                   FQUE1   *E1     L
*T EVENTR   B          T E1       1     F
*                      T E2       2     I
*                      T E3       4     I
*                      T E4       5     I
```

```
+                                  T  E5      6      F
+                                  T  E6      7      F
+                                  T  E7      8      F
+                                  T  E8     31      I
+                                  T  E9     32      I
+                                  T  E10    33      I
+                                  T  E11    34      I
+                                  T  E12    35      I
+                                  T  E13    36      F
+                                  T  E14    37      I
+                                  T  PEQUE 381/2    I
+                                  T  SEQUE 382/2    I
+
+T PH       8    8                 T  P1      1      F                              PQUE1   *P1      L
+                                  T  P2      2      I
+                                  T  P3      4      I
+                                  T  P4      5      I
+                                  T  P5      6      I
+                                  T  P6      7      I
+                                  T  PPQUE   8      I
+                                  T  SPQUE  31      I
+
+T OH       8    8                 T  H1      1      F                              HQUE1   *H1      L
+                                  T  H2      2      I
+                                  T  H3      4      I
+                                  T  H4      5      I
+                                  T  H5      6      I
+                                  T  H6      7      I
+                                  T  PHQUE   8      I
                                   T  SHQUE  31      I
```

```
*IBFTC MAIN
              MAIN ROUTINE
C
C
C        .....PLANET - WEAPON SYSTEM AVAILABILITY.....
C
C
C.....PURPOSE - TO REPORT OFF-ALERT STATUS.
C
C
C.....INPUT   - TAPE FROM ABC MODEL.
C
C
C.....OUTPUT  - PRINTER (SIMSCRIPT RPG).
C
C
C
      CALL HDING
C
      REWIND 9
    1 CALL RLBL
      CALL SELECT
C
      IF (IDSUB) EQ (   3), GO TO 3
      IF (BFLAG) NE (   0), GO TO 9999
      IF (IDSUB) EQ ( 100), GO TO 100
      IF (IDSUB) EQ ( 110), GO TO 110
      IF (IDSUB) EQ ( 200), GO TO 200
      IF (IDSUB) EQ ( 500), GO TO 500
      IF (IDSUB) EQ ( 600), GO TO 600
      IF (IDSUB) EQ ( 800), GO TO 800
      IF (IDSUB) EQ ( 900), GO TO 900
      IF (IDSUB) EQ ( 925), GO TO 925
      IF (IDSUB) EQ ( 950), GO TO 950
      IF (IDSUB) EQ (1900), GO TO 1900
      IF (IDSUB) EQ (2000), GO TO 2000
      IF (IDSUB) EQ (2100), GO TO 2100
      IF (IDSUB) EQ (2150), GO TO 2150
      IF (IDSUB) EQ (2300), GO TO 2300
      IF (IDSUB) EQ (2400), GO TO 2400
      IF (IDSUB) EQ (2500), GO TO 2500
C
      GO TO 9999
C
    3 CALL R3
      CALL EXIT
C
  100 CALL R100
      GO TO 9999
C
  110 CALL R110
      GO TO 9999
C
```

```
  200 CALL R200
      GO TO 9999
C
  500 CALL R500
      GO TO 9999
C
  600 CALL R600
      GO TO 9999
C
  800 CALL R800
      GO TO 9999
C
  900 CALL R900
      GO TO 9999
C
  925 CALL R925
      GO TO 9999
C
  950 CALL R950
      GO TO 9999
C
 1900 CALL R1900
      GO TO 9999
C
 2000 CALL R2000
      GO TO 9999
C
 2100 CALL R2100
      GO TO 9999
C
 2150 CALL R2150
      GO TO 9999
C
 2300 CALL R2300
      GO TO 9999
C
 2400 CALL R2400
      GO TO 9999
C
 2500 CALL R2500
      GO TO 9999
C
 9999 IF (MORE) EQ (0), GO TO 1
      CALL RDTL
      GO TO 9999
C
      END
```

```
*IBFTC RLBL
           SUBROUTINE RLBL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....LABEL RECORDS.
C
      LET BFLAG = 0
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
C
      STORE I1 IN IDSUB
      STORE I2 IN IDSUB
      STORE I3 IN SHFT
      STORE I4 IN DAYW
      STORE I5 IN SXOW
      STORE I6 IN EBAS
      STORE I7 IN VA
      STORE I8 IN VB
      STORE I9 IN VC
      STORE I10 IN TRSM
      STORE I11 IN MORE
      STORE I12 IN ETIME
C
      IF (ETIME) GR (TMEND), GO TO 1
C
      RETURN
C
    1 CALL R3
      CALL EXIT
C
      END
```

```
*IBFTC RDTL
               SUBROUTINE RDTL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....DETAIL RECORDS.
C
X      READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
C
       STORE I1 IN DTLV1
       STORE I2 IN DTLV2
       STORE I3 IN DTLV3
       STORE I4 IN DTLV4
       STORE I5 IN DTLV5
       STORE I6 IN DTLV6
       STORE I7 IN DTLV7
       STORE I8 IN DTLV8
       STORE I9 IN DTLV9
       STORE I10 IN DTLVC
C
       LET MORE = MORE - 1
C
       RETURN
       END
```

```
*IBFTC SELECT
               SUBROUTINE SELECT
C
C
C.....PURPOSE - TO SELECT EVENTS BY BASE.
C
C
       DO TO 1, FOR EACH BASES I
       IF (EBAS) EQ (BASE(I)), GO TO 2
     1 LOOP
       LET BFLAG = 1
       GO TO 9999
C
     2 LET BFLAG = 0
       GO TO 9999
C
  9999 RETURN
       END
```

```
*IBFTC R3
            SUBPOUTINE R3
C
C
C.....PURPOSE - TO CLOSE-OUT AND END R-PHASE.
C
C
C.....IDSUB = 3
C
C
      CALL HDING
C
      LET ILINES = 0
C
    1 DO TO 2, FOR EACH IS IN SQUF
      CALL REPORT(IS,ILINES)
    2 REPEAT 1
C
      CALL PQHD
C
    3 DO TO 6, FOR EACH IS IN SQUE
    4 DO TO 5, FOR EACH IP IN PQUE(IS)
C
      CALL PORPG(IS,IP)
C
    5 REPEAT 4
    6 REPEAT 3
C
      CALL HQHD
C
    7 DO TO 10, FOR EACH IS IN SQUE
    8 DO TO  9, FOR EACH IH IN HQUE(IS)
C
      CALL HORPG(IS,IH)
C
    9 REPEAT 8
   10 REPEAT 7
C
      REWIND 9
C
      RETURN
      END
```

```
*IBFTC R100
                SUBROUTINE R100
C
C
C.....PURPOSE - ALERT DEMAND ARRIVAL AT BASE CONTROL.
C
C
C.....IDSUB = 100.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL ROTL
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (DTLV1),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E9(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(103) IN E2(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R110
              SUBROUTINE R110
C
C
C.....PURPOSE - EXOG. FAILURE, EXOG. PM, EXOG. OVERHAUL.
C
C.....IDSUB = 110.
C
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (VC) EQ (2), GO TO 2
      IF (VC) EQ (4), GO TO 4
      IF (VC) EQ (6), GO TO 6
      GO TO 9999
C
C..........EXOG. FAILURE.
C
    2 DO TO 20, FOR EACH FLVLS I
      IF (VB) EQ (FLVL(I)), GO TO 21
   20 LOOP
      GO TO 9999
C
   21 LET NFAIL(IS) = NFAIL(IS) + 1
      CREATE EVENT CALLED E
      STORE ETIME IN E1(E)
      STORE VA    IN E3(E)
      STORE VB    IN E4(E)
      STORE TRSM  IN E10(E)
      STORE IDSUB IN E11(E)
      FILE E IN EQUE(IS)
      GO TO 9999
C
C..........EXOG. PM.
C
    4 CREATE PM CALLED P
      STORE ETIME IN P1(P)
      STORE VA    IN P2(P)
      STORE IDSUB IN P3(P)
      STORE TRSM  IN P5(P)
      FILE P IN PQUE(IS)
      GO TO 9999
C
C..........EXOG. OVERHAUL.
C
    6 CREATE OH CALLED H
      STORE ETIME IN H1(H)
      STORE VA    IN H2(H)
      STORE TRSM  IN H5(H)
      FILE H IN HQUE(IS)
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R200
              SUBROUTINE R200
C
C
C.....PURPOSE - MAINTENANCE COMPLETED.
C
C
C.....IDSUB = 200.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (DTLV1),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      CREATE EVENT CALLED E
C
      STORE ETIME      IN E1(E)
      STORE STAT(108)  IN E2(E)
      STORE E3(IE)     IN E3(E)
      STORE E4(IE)     IN E4(E)
      STORE IDSUB      IN E11(E)
C
      LET E8(IE) = 0
      LET E9(IE) = 0
      LET E10(IE) = 0
C
      LET E6(IE) = ETIME - E1(IE)
C
      FILE E IN EQUE(IS)
C
      LET NFAIL(IS) = NFAIL(IS) - 1
C
      IF (NFAIL(IS)) EQ (0), GO TO 99
      GO TO 9999
C
   99 CALL REPORT(IS,ILINES)
C
 9999 RETURN
      END
```

```
*IBFTC R500
            SUBROUTINE R500
C
C
C.....PURPOSE - TEAM DISPATCH BY BASE CONTROL.
C
C
C.....IDSUB = 500.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      IF (DTLV4) EQ (MSITE), GO TO 9999
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (DTLV1),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E9(N)) EQ (VC),
     XWHERE IE, IF NONE, GO TO 1
C
      CREATE EVENT CALLED E
C
      STORE DTLV9  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE TRSM   IN E8(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(104) IN E2(E)
C
      FILE E IN EQUE(IS)
C
      STORE TRSM IN E8(IE)
C
      GO TO 9999
C
    1 FIND FIRST, FOR EACH M IN HQUE(IS), WITH (H4(M)) EQ (VC),
     XWHERE IH, IF NONE, GO TO 9999
C
      STORE TRSM IN H3(IH)
C
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R600
            SUBROUTINE R600
C
C
C.....PURPOSE - TEAM ARRIVAL AT SITE.
C
C
C.....IDSUB = 600.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (VB) EQ (6), GO TO 6
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XAND (E11(N)) EQ (500), WHERE IE, IF NONE, GO TO 9999
C
      LET TLAG = ETIME - E1(IE)
C
      LET E8(IE) = 0
      LET E9(IE) = 0
      LET E10(IE) = 0
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      LET E5(IE) = E5(IE) + TLAG
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(105) IN E2(E)
C
      FILE E IN EQUE(IS)
C
      GO TO 9999
C
    6 FIND FIRST, FOR EACH M IN HQUE(IS), WITH (H3(M)) EQ (TRSM),
     XWHERE IH, IF NONE, GO TO 9999
C
      LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE H6(IH) IN E2(E)
      STORE H2(IH) IN E3(E)
      STORE TRSM   IN E8(E)
      STORE H4(IH) IN E9(E)
```

```
      STORE H5(IH) IN E10(E)
      LET E11(E) = 110
C
      FILE E IN EQUE(IS)
C
      REMOVE IH FROM HQUE(IS)
      DESTROY OH CALLED IH
C
      GO TO 9999
C
 9999 RETURN
      END



*IBFTC R800
            SUBROUTINE R800
C
C
C.....PURPOSE - TEAM LOST ENROUTE TO BASE.
C
C
C.....IDSUB = 800.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XAND (E11(N)) EQ (500), WHERE IE, IF NONE, GO TO 9999
C
      LET TLAG = ETIME - E1(IE)
C
      LET E8(IE) = 0
      LET E9(IE) = 0
      LET E10(IE) = 0
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      LET E5(IE) = E5(IE) + TLAG
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(106) IN E2(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R900
              SUBROUTINE R900
C
C
C.....PURPOSE - EQUIP. FAILURE AT SITE.
C
C
C.....IDSUB = 900.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(107) IN E2(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END

*IBFTC R925
              SUBROUTINE R925
C
C
C.....PURPOSE - PERS. FAILURE AT SITE.
C
C
C.....IDSUB = 925.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(111) IN E2(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R950
              SUBROUTINE R950
C
C
C.....PURPOSE - PARTS FAILURE AT SITE.
C
C
C.....IDSUB = 950.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E8(N)) EQ (TRSM),
     XWHERE IE, IF NONE, GO TO 9999
C
      CREATE EVENT CALLED E
C
      STORE FTIME  IN E1(E)
      STORE E3(IE) IN E3(E)
      STORE E4(IE) IN E4(E)
      STORE IDSUB  IN E11(E)
      STORE STAT(112) IN E2(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R1900
              SUBROUTINE R1900
C
C
C.....PURPOSE - GENERATE SITES AT BASE.
C
C
C.....IDSUB = 1900.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      CREATE SITES CALLED S
C
      LET I = I + 1
C
      STORE TRSM   IN SID(S)
      STORE VC     IN SMODE(S)
      STORE DTLV1  IN SERNO(S)
      STORE EBAS   IN EBASE(S)
      STORE I      IN SINO(S)
C
      IF (VC) GE (MSITE), LET MSITE = VC + 1
C
      FILE S IN SQUE
C
 9999 RETURN
      END


*IBFTC R2000
              SUBROUTINE R2000
C
C
C.....PURPOSE - REQUEST FOR PM.
C
C
C.....IDSUB = 2000.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      CREATE PM CALLED P
C
      STORE FTIME  IN P1(P)
      STORE VB     IN P2(P)
      STORE TRSM   IN P5(P)
C
      FILE P IN PQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R2100
            SUBROUTINE R2100
C
C
C.....PURPOSE - BEGIN PM.
C
C
C.....IDSUB = 2100.
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN PQUE(IS), WITH (P5(N)) EQ (TRSM),
     XAND (P2(N)) EQ (VB), WHERE IP, IF NONE, GO TO 9999
C
      LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE EVENT CALLED E
C
      STORE ETIME  IN E1(E)
      STORE P6(IP) IN E2(E)
      STORE P2(IP) IN E3(E)
      STORE VC     IN E8(E)
      STORE P4(IP) IN E9(E)
      STORE P5(IP) IN E10(E)
      STORE IDSUB  IN E11(E)
C
      IF (P3(IP)) EQ (110), GO TO 110
      GO TO 9998
C
  110 STORE P3(IP) IN E11(E)
      GO TO 9998
C
 9998 FILE E IN EQUE(IS)
C
      REMOVE IP FROM PQUE(IS)
      DESTROY PM CALLED IP
C
 9999 RETURN
      END
```

```
*IBFTC R2150
            SUBROUTINE R2150
C
C
C.....PURPOSE - FAILURE CAUSED BY PM.
C
C
C.....IDSUB = 2150.
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH M IN SQUF, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE EVENT CALLED E
C
      STORE ETIME IN E1(E)
      STORE VB    IN E3(E)
      STORE VA    IN E4(E)
      STORE TRSM  IN E10(E)
      STORE IDSUB IN E11(E)
C
      FILE E IN EQUF(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R2300
              SUBROUTINE R2300
C
C
C.....PURPOSE - ALERT - CONTINUOUS MONITOR.
C
C
C.....IDSUB = 2300.
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE EVENT CALLED E
C
      STORE ETIME IN E1(E)
      STORE VB    IN E3(E)
      STORE VA    IN E4(E)
      STORE TRSM  IN E10(E)
      STORE IDSUB IN E11(E)
C
      FILE E IN EQUE(IS)
C
 9999 RETURN
      END
```

```
*IBFTC R2400
              SUBROUTINE R2400
C
C
C.....PURPOSE - RESOURCE REQUEST.
C
C
C.....IDSUB = 2400.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
      LET DTLV2 = DTLV2 + 1
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (DTLV1) EQ (1), GO TO 1
      IF (DTLV1) EQ (2), GO TO 2
      IF (DTLV1) EQ (3), GO TO 3
      IF (DTLV1) EQ (4), GO TO 4
      IF (DTLV1) EQ (5), GO TO 5
      IF (DTLV1) EQ (6), GO TO 6
      GO TO 9999
C
    1 FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E11(N)) EQ (2300),
     XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
     XGO TO 9999
C
      STORE STAT(101) IN E2(IE)
      STORE VC        IN E9(IE)
      GO TO 9999
C
    2 FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E11(N)) EQ (110),
     XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
     XGO TO 9999
C
      STORE STAT(109) IN E2(IE)
      STORE VC        IN E9(IE)
      GO TO 9999
C
    3 FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E11(N)) EQ (2500),
     XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
     XGO TO 9999
C
      STORE STAT(102) IN E2(IE)
      STORE VC        IN E9(IE)
C
      CREATE EVENT CALLED E
C
      STORE ETIME        IN E1(E)
      STORE STAT(DTLV2)  IN E2(E)
      STORE VB           IN E3(E)
```

```
      STORE VA              IN E4(F)
      STORE IDSUB           IN E11(F)
C
      FILE E IN EQUE(IS)
C
      GO TO 9999
C
    4 FIND FIRST, FOR EACH N IN PQUE(IS), WITH (P5(N)) EQ (TRSM),
     XAND (P2(N)) EQ (VB), WHERE IP, IF NONE, GO TO 9999
C
      STORE VC              IN P4(IP)
      LET VB = VB + 1
      STORE STAT(VB)        IN P6(IP)
      GO TO 9999
C
    5 FIND FIRST, FOR EACH N IN EQUE(IS), WITH (E11(N)) EQ (2150),
     XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
     XGO TO 9999
C
      STORE STAT(DTLV2) IN E2(IE)
      STORE VC              IN E9(IE)
      GO TO 9999
C
    6 FIND FIRST, FOR EACH N IN HQUE(IS), WITH (H5(N)) EQ (TRSM),
     XAND (H2(N)) EQ (VB), WHERE IH, IF NONE, GO TO 9999
C
      STORE VC          IN H4(IH)
      STORE STAT(110) IN H6(IH)
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R2500
              SUBROUTINE R2500
C
C
C.....PURPOSE - UNDETERMINED FAILURE.
C
C
C.....IDSUB = 2500.
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE EVENT CALLED E
C
      STORE ETIME IN E1(E)
      STORE VB    IN E3(E)
      STORE VA    IN E4(E)
      STORE TRSM  IN E10(E)
      STORE IDSUB IN E11(E)
C
      FILE E IN EQUF(IS)
C
 9999 RETURN
      END
```

```
*IDFTC REPORT
              SUBROUTINE REPORT(IS,ILINES)
C
C
C.....PURPOSE - TO REPORT THE EVENTS OF A SITE FOR OFF-ALERT STATUS.
C
C
C.....CALLED BY R200 OR R3.
C
C
      IF EQUE(IS) IS EMPTY, GO TO 9999
C
      STORE E2(FEQUE(IS)) IN E12(FEQUE(IS))
      LET E13(FEQUE(IS)) = E1(LFQUE(IS)) - E1(FEQUE(IS))
C
    1 DO TO 2, FOR EACH IE IN EQUE(IS)
      LET ISEQ = ISEQ + 1
      LET E14(IE) = ISEQ
      IF (E12(IE)) EQ (0), LET E12(IE) = STAT(1)
    2 REPEAT 1
      LET ISEQ = 0
C
      CALL ALDEG(IS)
C
   10 IF EQUE(IS) IS EMPTY, GO TO 9998
      LET IE = FEQUE(IS)
C
      LET CTIME = E1(IE)
      CALL CNVRT(CTIME,ID,IH,IM)
      LET ID1 = ID
      LET IH1 = IH
      LET IM1 = IM
C
      LET CTIME = E13(IE)
      CALL CNVRT(CTIME,ID,IH,IM)
      LET ID2 = ID
      LET IH2 = IH
      LET IM2 = IM
C
      LET CTIME = E5(IE)
      CALL CNVRT(CTIME,ID,IH,IM)
      LET ID3 = ID
      LET IH3 = IH
      LET IM3 = IM
C
      LET CTIME = E6(IE)
      CALL CNVRT(CTIME,ID,IH,IM)
      LET ID4 = ID
      LET IH4 = IH
      LET IM4 = IM
C
      LET CTIME = E7(IE)
      CALL CNVRT(CTIME,ID,IH,IM)
```

```
      LET ID5 = ID
      LET IH5 = IH
      LET IM5 = IM
C
      LET ILINES = ILINES + 1
      IF (ILINES) LS (55), GO TO 20
      CALL HDING
      LET ILINES = 0
C
   20 CALL FORM(IE,IS,ID1,IH1,IM1,ID2,IH2,IM2,ID3,IH3,IM3,ID4,IH4,IM4,
     XID5,IH5,IM5)
C
      REMOVE IE FROM EQUE(IS)
      DESTROY EVENT CALLED IE
      GO TO 10
C
 9998 CALL BLANK
      LET ILINES = ILINES + 1
C
 9999 RETURN
      END
```

```
*IBFTC ALDEG
              SUBROUTINE ALDEG(IS)
C
C
C.....PURPOSE - TO CALCULATE ALERT DEGRADATION TIME FOR A SITE.
C
C
C.....CALLED BY REPORT.
C
C
      LET LFAIL = 0
      LET LFIX  = 0
      LET PTIME = 0.0
      LET IE14 = 0
C
    1 DO TO 2, FOR EACH IE IN EQUE(IS), WITH (E14(IE)) NE (0)
      IF (E11(IE)) EQ (2500), GO TO 3
      IF (E11(IE)) EQ (2100), GO TO 3
      IF (E11(IE)) EQ (2300), GO TO 3
      IF (E11(IE)) EQ (2150), GO TO 3
      IF (E11(IE)) EQ ( 110), GO TO 3
      LET E14(IE) = 0
    2 REPEAT 1
C
      GO TO 999
C
    3 LET IFAIL = IE
      FIND FIRST, FOR EACH M IN EQUE(IS), WITH (E11(M)) EQ ( 200), AND
     X(E3(M)) EQ (E3(IFAIL)), AND (E4(M)) EQ (E4(IFAIL)), AND (E14(M))
     XGR (LFIX), WHERE I200, IF NONE, GO TO 40
C
      IF (LFAIL) EQ (0), GO TO 20
      LET E7(IFAIL) = E1(I200) - PTIME
      GO TO 30
C
   20 LET E7(IFAIL) = E1(I200) - E1(IFAIL)
C
   30 LET LFIX = E14(I200)
      LET PTIME = E1(I200)
C
   40 LET IFAIL = E14(IFAIL
      LET E14(IE) = 0
      GO TO 2
C
  999 DO TO 9999, FOR EACH IE IN EQUE(IS)
      LET IE14 = IE14 + 1
      LET E14(IE) = IE14
 9999 REPEAT 999
C
      RETURN
      END
```

```
*IBFTC CNVRT
            SUBROUTINE CNVRT(CTIME,ID,IH,IM)
C
C
C.....PURPOSE - TO CONVERT DECIMAL DAYS TO DAYS(ID), HOURS(IH), MINS(IM)
C
C
C.....CALLED BY REPORT.
C
C
       LET ID = DPART(CTIME)
       LET IH = HPART(CTIME)
       LET IM = MPART(CTIME)
       IF (IM) NE (60), GO TO 1
       LET IH = IH + 1
       LET IM = 0
     1 IF (IH) LS (24), GO TO 9999
       LET IH = IH - 24
       LET ID = ID + 1
C
 9999 RETURN
       END
```

```
*IBFTC HDING
       REPORT HDING
*             EVENT  FAIL  SITE          INIT       TOTAL     EVNT  UNIT  FAI
*              TIME   NO   TYPE  SQUAD   STAT    OFF-ALERT    STAT  FAIL  LEV
              END
                                                                              *
L         LAG          DUR.      ALERT     SEC
L         TIME         TIME   DEG. TIME    NO.    TID    PID    SID EVNT
       END
```

```
*IBFTC FORM
       REPORT FORM(IE,IS,ID1,IH1,IM1,ID2,IH2,IM2,ID3,IH3,IM3,ID4,IH4,IM4,
*        * * *    *     *     *   *,       * * *    *A      *
 *     ID1,IH1,IM1,SERNO(IS),SMODE(IS),EBASE(IS),E12(IE),ID2,IH2,IM2,E2(I
 *     E8(IE),E9(IE),E10(IE),E11(IE)
       END
ID5,IH5,IM5)
*    * * *    * * *    * * *    *    *    *    *    *    *
E),E3(IE),E4(IE),ID3,IH3,IM3,ID4,IH4,IM4,ID5,IH5,IM5, E16(IE),
                                                                              *
       END
```

```
*IBFTC BLANK
      REPORT BLANK
*
      END
                                                                      *
                                                                    *
      END



*IBFTC PQHD
      REPORT PQHD
*      PM EVENT  TAIL  EVNT  UNIT
*          TIME   NO   STAT  FAIL          RID   SID
      END
                                                                      *
                                                                      *
                                                                  1
      END


*IBFTC PQRPG
      REPORT PQRPG(IS,IP)
*     ***.*****     *    *A     *          *    *
 *    P1(IP),SERNO(IS),P6(IP),P2(IP),P4(IP),P5(IP)
      END
                                                                      *
                                                                    *
                                                                      *

      END



*IBFTC HQHD
      REPORT HQHD
*      OH EVENT  TAIL  EVNT  UNIT
*          TIME   NO   STAT  FAIL   TID   RID   SID
      END
                                                                      *
                                                                      *
                                                                  1
      END


*IBFTC HQRPG
      REPORT HQRPG(IS,IH)
*     ***.*****     *    *A     *     *     *     *
 *    H1(IH),SERNO(IS),H6(IH),H2(IH),H3(IH),H4(IH),H5(IH)
      END
                                                                      *
                                                                    *
                                                                      *

      END
```

```
*ENTRY          MAIN
1           33
   1  22 0   Z                                          99
  23     0 R
  24     1 R   100  23                                    100
00                                                        (12)
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47           P
```

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
25 0 2

26 0 R
27 1 R 10 26 10
(I1)
0
1
2
3
4
5
6
7
8
9
28 0 R
29 1 R 200 28 200
(A2)
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

92
93
94
95
96
97
98
99
AA
AB
AD
AT
AS
LO
EE
XA
ZZ
OH
FP
FS

30 32 0 7
33 0 8

999.99999

Program 3

LOGISTICS RESOURCE UTILIZATION

III. LOGISTICS RESOURCE UTILIZATION

The logistics information is presented in three parts: Spares, Personnel, and Equipment. The spare parts data presented by this report contain information regarding stock levels, NORS (not operationally ready--supply) time, NORS count, and demand quantities. The personnel report contains the man-hour accounting information. This report displays the utilization factors for each personnel type and the man-hours consumed by various tasks. The equipment data presented contain information regarding the utilization of maintenance equipment and facilities. NORE (not operationally ready--equipment) time, NORE count, and demand quantities are included.

The Logistics Resource Utilization report can be used for either aircraft or missile simulations.

Fig. 11 is a display of some example data showing the spare parts usage for the period* ending day 21.00 of the simulation. Column 1 lists each part type specified in the input data of the simulator. Column 2 lists the authorized quantity, also specified in the inputs of the simulator. Column 3 lists, for each spare type, the quantity of serviceable items remaining in stock at report time. Column 4 lists the number of demands for each spare part during the reporting period. Column 5 is the total demands for each spare part. Column 6 is the NORS (not operationally ready--supply) time during the reporting period. NORS time in this report is treated independently of the other NOR conditions, i.e., this is the maximum amount of time that could be attributed to Supply. Column 7 is the accumulation of all NORS time to date. Column 8 lists, for each spare part, the number of stockouts that occurred during the period. Column 9 lists the total number of stockouts accumulated to date.

Figure 12 is a display of some example data showing the man-hour accounting information for the period* ending day 21.00 of the simulation. Column 1 lists each personnel type specified in the input data of the simulator. Column 2 is the man-shifts (number of personnel x number of

*The period is an initialized variable.

SPARE PARTS USAGE FOR PERIOD ENDING 21.00

TYPE	AUTH. QTY	TOTAL QTY AVAILABLE	SITE DEMANDS THIS PERIOD	SITE DEMANDS TO DATE	NORS TIME THIS PERIOD	NORS TIME TO DATE	NO. NORS THIS PER	NO. NORS TO DATE
1	50	18	10	40	0.09	0.09	6	6
2	20	5	8	21	35.21	35.26	52	53
3	16	16	40	138	0.00	0.00	0	0
4	40	34	9	29	0.00	0.00	0	0
5	10	10	0	0	0.00	0.00	0	0

Fig. 11 -- Spare Parts Usage

MANHOUR ACCOUNTING FOR PERIOD ENDING 21.00

TYPE	MAN SHFT AVAIL	TOTAL MANHRS THIS PERIOD	TOTAL MANHRS TO DATE	FLT LINE MAINT HRS	INTRAN HOURS	OVERTIME HOURS	UTIL FACTOR THIS PERIOD	UTIL FACTOR TO DATE
1	1052	8416.00	25232.00	8.06	2.73	0.00	0.00	0.00
2	422	3376.00	10112.00	1.17	0.01	0.00	0.00	0.00
3	338	2704.00	8096.00	7.56	0.31	1.01	0.00	0.00
4	842	6736.00	20192.00	15.53	3.09	0.00	0.00	0.00
5	212	1696.00	5072.00	14.12	0.47	0.81	0.01	0.01
6	420	3360.00	10080.00	9.98	0.42	0.00	0.00	0.00

Fig. 12 -- Man-hour Accounting

MAINT. EQUIPMENT USAGE FOR PERIOD ENDING 21.00

TYPE	AUTH. QTY	TOTAL QTY AVAILABLE	SITE DEMANDS THIS PERIOD	SITE DEMANDS TO DATE	NORE TIME THIS PERIOD	NORE TIME TO DATE	NO. NORE THIS PER	NO. NORE TO DATE
1	50	50	52	218	0.00	0.00	0	0
2	20	20	18	61	0.00	0.00	0	0
3	16	16	8	2[illegible]	0.00	0.00	0	0
4	40	40	1	5	0.00	0.00	0	0

Fig. 13 -- Maintenance Equipment Usage

shifts that the personnel are assigned) available. Column 3 lists, for each personnel type, the total man-hours (man-shifts x shift duration) available. Column 4 lists the accumulation of the total man-hours available to date. Column 5 lists the maintenance man-hours (decimal hours) for each personnel type. Column 6 lists, for each personnel type, the man-hours spent in transit to and from the site. Column 7 lists the overtime required. Column 8 displays the utilization of each personnel type (Σ Col. 5,6,7 $\div$ Col. 3). Column 9 is the accumulation of the utilization factor for the simulation to date (Σ all Col. 5,6, and 7's $\div$ Col. 4).

Figure 13 is a display of some example data showing the maintenance equipment usage for the period* ending day 21.00. Column 1 lists each equipment type specified in the input data of the simulator. Column 2 lists the authorized quantity, also specified in the inputs of the simulator. Column 3 lists, for each equipment type, the quantity available for use at report time. Column 4 lists the number of demands for each equipment type during the reporting period. Column 5 is the total demands for each equipment type. Column 6 is the NORE (not operationally ready--equipment) time during the reporting period. NORE time is treated independently of other NOR conditions, i.e., this is the maximum amount of time that could be attributed to Equipment if there were no other NOR conditions. Column 7 is the accumulation of all NORE time to date. Column 8 lists, for each equipment type, the number of demands that could not be satisfied immediately because of a shortage of equipment. Column 9 lists the total number of NORE conditions to date.

INITIALIZATION

Table 3 lists the initialization requirements. Only six arrays require inputs. Array 23 is the report interval for the man-hour accounting report. This report interval must be initialized either daily or weekly. Arrays 24 and 25 are the reporting interval for Spares and Equipment and can be initialized to any floating point (decimal)

*The period is an initialized variable.

Table 3

VARIABLE DESCRIPTION AND INITIALIZATION:
LOGISTICS RESOURCE UTILIZATION

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col.: 19-22 (base)	Array Number of Attribute to Be Entered in Fig. 5 Col.: 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1-22	0			X										
23	0		F		V						Reporting interval for manhour accounting (Must be 1.00000 or 7.00000)	T10	X	
24	0		F		V						Reporting interval for maintenance equipment usage.	T20	X	
25	0		F		V						Reporting interval for spare parts usage	T30	X	
26-28	0			X										A
29	0		F		V						Hours per shift (8.00000)	SHIFT	X	
30-32	0			X										A
33	0	I			V						Number of bases to be analyzed	BASES	X	
34	1	I			V			33			Specify each base number to be analyzed	BASE		A
35-38	0			X										

number. Array 29 specifies the duration of the work shift (floating point) for use with the man-hour accounting report. Arrays 33 and 34 specify the number of bases to be contained in the report and the bases(s) number(s). The remaining arrays are zeroed.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of one or more bases may be run at one time.

When a label record is read, ETIME is compared to RT10, RT20, and RT30. If ETIME is greater than or equal to RT10, RT20, or RT30, then RPT10, RPT20, and/or RPT30 is called and the reports are displayed.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-PR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/wk
SXDW = S/wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRSM = ID Addresses
MORK = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-PR).

DTLV1 = Integer variable 1

DTLV2 = Integer variable 2

DTLV3 = Integer variable 3

DTLV4 = Integer variable 4

DTLV5 = Integer variable 5

DTLV6 = Integer variable 6

DTLV7 = Integer variable 7

DTLV8 = Integer variable 8

DTLV9 = Float variable 1

DTLV0 = Float variable 2

T10 = RT10 increments for Personnel display.
Must be 1.00000 or 7.00000.

T20 = RT20 increments for equipment display.
No restrictions.

T30 = RT30 increments for spare parts display.
No restrictions.

SHIFT = Hours per shift.

BASES = Number of base codes to be processed.

BASE = Base codes to be processed.

BFLAG = Controls flow of events as a result of EBAS vs. BASES.

<u>SETS</u>

Name = LQ10 used for display of Personnel. No subscripts.
Ranked on L11.

Owner = SIMSCRIPT system.

Member = L10 (Time is in decimal hours).

L11 = Type.

L12 = Man-shift available.

L13 = Total man-hours this period.

L14 = Total man-hours to date.

L15 = Flight-line maintenance hours this period.

L15A = Flight-line maintenance hours to date.

L16 = Not used.

L17 = Intransit hours for this period.

L17A = Intransit hours to date.

L18 = Overtime hours for this period.

L18A = Overtime hours to date.

L19 = Utilization factor for this period.

L110 = Utilization factor to date.

Name = LQ20 used for display of equipment. No subscripts. Ranked on L21.

Owner = SIMSCRIPT system.

Member = L20 (Time is in decimal days).

L21 = Type.

L22 = Authorized quantity.

L23 = Total quantity available.

L24 = Site demands this period.

L25 = Site demands to date.

L26 = NORE time this period.

L27 = NORE time to date.

L28 = Number of NORE this period.

L29 = Number of NORE to date.

Name = LQ30 used for display of spares. No subscripts. Ranked on L31.

Owner = SIMSCRIPT system.

Member = L30 (Time is in decimal days).

L31 = Type.

L32 = Authorized quantity.

L33 = Total quantity available.

L34 = Site demands this period.

L35 = Site demands to date.

L36 = NORS time this period.

L37 = NORS time to date.

L38 = Number of NORS this period.

L39 = Number of NORS to date.

Name = PQUE used to queue personnel. No subscripts. Ranked on TP10.

Owner = SIMSCRIPT system.

Member = P10.

TP10 = Type.

SP10 = Shift.

QP10 = Quantity.

DP10 = Day.

PFLG = Control flag to compute day.

Name = PQ12 used to queue exogenous personnel. No subscripts. Ranked on TP12.

Owner = SIMSCRIPT system.

Member = P12.

TP12 = Type.

SP12 = Shift

QP12 = Quantity.

DP12 = Day.

Name = BQUE used for team resources. No subscripts. Ranked on RTID. RQUE used for team resources. One subscript. Ranked on TTID.

Owner = BQUE is SIMSCRIPT system.
RQUE is Team.

Member = RESRC.

RTID = Team ID.

RRID = Request ID.

RTYP = Type.

RQTY = Quantity.

RSUB = Resource.

RSFT = Shift.

Name = TQUE used to queue teams. No subscripts. Ranked on T500.

Owner = SIMSCRIPT system.

Member = TEAM.

TTID = Team ID.

TRID = Request ID.

T500 = Time team left base/site.

T504 = Time team left site.

T600 = Time team arrived at site.

T700 = Time team arrived at base.

T200 = Time maintenance was completed.

T1450 = Time team was returned to base pool.

T800 = Time team lost (if applicable).

S500 = Time team left Base.

Name = NQUE used for NORS and NC'E. No subscripts. Ranked on TNOR.

Owner = SIMSCRIPT system.

Member = NOR.

NTID = Team ID.

NRID = Request ID.

NTYP = Type.

NQTY = Quantity.

NSUB = NORS or NCRE.

TNOR = Time of NORS or NORE.

```
♦                                     11DSUB  0    I
♦                                     21DSUB  0    I
♦                                     3SHFT   0    I
♦                                     4DAYW   0    I
♦                                     5SXDW   0    I
♦                                     6FBAS   0    I
♦                                     7VA     0    I
♦                                     8VB     0    I
♦                                     9VC     0    I
♦                                     10TRSM  0    I
♦                                     11MORE  0    I
♦                                     12FTIME 0    F
♦                                     13DTLV1 0    I
♦                                     14DTLV2 0    I
♦                                     15DTLV3 0    I
♦                                     16DTLV4 0    I
♦                                     17DTLV5 0    I
♦                                     18DTLV6 0    I
♦                                     19DTLV7 0    I
♦                                     20DTLV8 0    I
♦                                     21DTLV9 0    F
♦                                     22DTLV0 0    F
♦                                     23T10   0    F
♦                                     24T20   0    F
♦                                     25T30   0    F
♦                                     26RT10  0    F
♦                                     27RT20  0    F
♦                                     28RT30  0    F
♦                                     29SHIFT 0    F
♦                                     30MAX10 0    I
♦                                     31MSITF 0    I
♦                                     32BFLAG 0    I
♦                                     33BASES E    I
♦                                     34BASE  1    I
♦                                     35FBQUE 0    I
♦                                     36LBQUE 0    I
♦                                     37FTQUE 0    I
♦                                     38LTQUE 0    I
♦                                     39FPQUE 0    I
♦                                     40LPQUE 0    I
♦                                     41FLQ10 0    I
♦                                     42LLQ10 0    I
♦                                     43FLQ20 0    I
♦                                     44LLQ20 0    I
♦                                     45FLQ30 0    I
♦                                     46LLQ30 0    I
♦                                     47FNQUE 0    I
♦                                     48LNQUE 0    I
♦                                     49FPQ12 0    I
♦                                     50LPQ12 0    I
♦
♦T L10   8   8                                          LQ100   *L11    L
♦                 T L11    1    I
```

```
+                           T L12    2    I
+                           T L13    4    I
+                           T L14    5    F
+                           T L15    6    F
+                           T L15A   7    F
+                           T L16    8    I
+                           T L17   31    F
+                           T L17A  32    F
+                           T L18   33    F
+                           T L18A  34    F
+                           T L19   35    I
+                           T L110  36    F
+                           T PLQ1037     I
+                           T SLQ1038     I
+
+T L20   8   8                                        LQ200   *L21    L
+                           T L21    1    I
+                           T L22    2    I
+                           T L23    4    I
+                           T L24    5    I
+                           T L25    6    I
+                           T L26    7    F
+                           T L27    8    F
+                           T L28   31    I
+                           T L29   32    I
+                           T PLQ2033     I
+                           T SLQ2034     I
+
+T L30   8   8                                        LQ300   *L31    L
+                           T L31    1    I
+                           T L32    2    I
+                           T L33    4    I
+                           T L34    5    I
+                           T L35    6    I
+                           T L36    7    F
+                           T L37    8    F
+                           T L38   31    I
+                           T L39   32    I
+                           T PLQ3033     I
+                           T SLQ3034     I
+
+T P10   8                                            PQUEO   *TP10   L
+                           T TP10   1    I
+                           T SP10   2    I
+                           T QP10   3    I
+                           T DP10   4    I
+                           T PPQUE  5    I
+                           T SPQUE  6    I
+                           T PFLG   7    I
+
+T P12   8                                            PQ120   *TP12   L
+                           T TP12   1    I
+                           T SP12   2    I
```

```
*                    T OP12  3     I
*                    T OP12  4     I
*                    T PPQ12 5     I
*                    T SPQ12 6     I
*                                                        BQUFO   *RTID  L
*                                                        RQUE1   *TTID  L
*T RESRC8   8
*                    T RTID  1     I
*                    T RRID  2     I
*                    T RTYP  4     I
*                    T RQTY  5     I
*                    T RSUB  6     I
*                    T PBQUE 7     I
*                    T SBQUF 8     I
*                    T PRQUE31     I
*                    T SRQUE32     I
*                    T RSFT 33     I
*                                                        TQUFO   *T500  L
*T TEAM 8   8
*                    T TTID  1     I
*                    T TRID  2     I
*                    T T500  4     F
*                    T T504  5     F
*                    T T600  6     F
*                    T T700  7     F
*                    T T200  8     F
*                    T T145031     F
*                    T T800 32     F
*                    T PTQUE33     I
*                    T STQUE34     I
*                    T FRQUE35     I
*                    T LRQUE36     I
*                    T S500 37     F
*                                                        NQUEO   *TNOR  L
*T NOR   8
*                    T NTID  1     I
*                    T NRID  2     I
*                    T NTYP  3     I
*                    T NQTY  4     I
*                    T NSUB  5     I
*                    T TNOR  6     F
*                    T PNQUE 7     I
*                    T SNQUE 8     I
```

```
*IBFTC MAIN
            MAIN ROUTINE
C
C
C      .....PLANET - LOGISTICS RESOURCE UTILIZATION.....
C
C
C.....PURPOSE - TO REPORT (1) MANHOUR ACCOUNTING.
C                         (2) MAINT. EQUIPMENT USAGE.
C                         (3) SPARE PARTS USAGE.
C
C
C.....INPUT   - TAPE FROM ABC MODEL.
C
C
C.....OUTPUT  - PRINTER (SIMSCRIPT RPG).
C
C
      REWIND 9
C
      LET RT10 = T10
      LET RT20 = T20
      LET RT30 = T30
C
    1 CALL RLBL
      CALL SELECT
C
      IF (IDSUB) EQ (   3), GO TO    3
      IF (BFLAG) NE (   0), GO TO 9999
      IF (IDSUB) EQ (  10), GO TO   10
      IF (IDSUB) EQ (  20), GO TO   20
      IF (IDSUB) EQ (  30), GO TO   30
      IF (IDSUB) EQ ( 200), GO TO  200
      IF (IDSUB) EQ ( 500), GO TO  500
      IF (IDSUB) EQ ( 600), GO TO  600
      IF (IDSUB) EQ ( 700), GO TO  700
      IF (IDSUB) EQ ( 800), GO TO  800
      IF (IDSUB) EQ (1002), GO TO 1002
      IF (IDSUB) EQ (1010), GO TO 1010
      IF (IDSUB) EQ (1012), GO TO 1012
      IF (IDSUB) EQ (1020), GO TO 1020
      IF (IDSUB) EQ (1022), GO TO 1022
      IF (IDSUB) EQ (1100), GO TO 1100
      IF (IDSUB) EQ (1200), GO TO 1200
      IF (IDSUB) EQ (1210), GO TO 1210
      IF (IDSUB) EQ (1220), GO TO 1220
      IF (IDSUB) EQ (1400), GO TO 1400
      IF (IDSUB) EQ (1401), GO TO 1401
      IF (IDSUB) EQ (1450), GO TO 1450
      IF (IDSUB) EQ (1470), GO TO 1470
      IF (IDSUB) EQ (1900), GO TO 1900
C
      GO TO 9999
```

```
C
    3 CALL R3
      CALL EXIT
C
   10 CALL R10
      GO TO 9999
C
   20 CALL R20
      GO TO 9999
C
   30 CALL R30
      GO TO 9999
C
  200 CALL R200
      GO TO 9999
C
  500 CALL R500
      GO TO 9999
C
  600 CALL R600
      GO TO 9999
C
  700 CALL R700
      GO TO 9999
C
  800 CALL R800
      GO TO 9999
C
 1002 CALL R1002
      GO TO 9999
C
 1010 CALL R1010
      GO TO 9999
C
 1012 CALL R1012
      GO TO 9999
C
 1020 CALL R1020
      GO TO 9999
C
 1022 CALL R1022
      GO TO 9999
C
 1100 CALL R1100
      GO TO 9999
C
 1200 CALL R1200
      GO TO 9999
C
 1210 CALL R1210
      GO TO 9999
C
 1220 CALL R1220
```

```
      GO TO 9999
C
 1400 CALL R1400
      GO TO 9999
C
 1401 CALL R1401
      GO TO 9999
C
 1450 CALL R1450
      GO TO 9999
C
 1470 CALL R1470
      GO TO 9999
C
 1900 CALL R1900
      GO TO 9999
C
 9999 IF (MORE) EQ (0), GO TO 1
      CALL ROTL
      GO TO 9999
C
      END
```

```
*IBFTC RLBL
            SUBROUTINE RLBL
C
C.....READS S-PHASE TAPE(9) (BIN MODE).....LABEL RECORDS.
C
      LET BFLAG = 0
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
C
      STORE I1  IN IDSOR
      STORE I2  IN IDSUB
      STORE I3  IN SHFT
      STORE I4  IN DAYW
      STORE I5  IN SXDW
      STORE I6  IN FBAS
      STORE I7  IN VA
      STORE I8  IN VB
      STORE I9  IN VC
      STORE I10 IN TRSM
      STORE I11 IN MORF
      STORE I12 IN ETIME
C
    1 IF (ETIME) GE (RT10), GO TO 10
    2 IF (ETIME) GE (RT20), GO TO 20
    3 IF (ETIME) GE (RT30), GO TO 30
C
      RETURN
C
   10 CALL RPT10
      GO TO 1
C
   20 CALL RPT20
      GO TO 2
C
   30 CALL RPT30
      GO TO 3
C
      END
```

```
*IBFTC SELECT
              SUBROUTINE SELECT
C
C
C.....PURPOSE - TO SELECT EVENTS BY BASE.
C
C
      DO TO 1, FOR EACH BASES I
      IF (FBAS) EQ (BASE(I)), GO TO 2
    1 LOOP
      LET BFLAG = 1
      GO TO 9999
C
    2 LET BFLAG = 0
      GO TO 9999
C
 9999 RETURN
      END



*IBFTC RDTL
              SUBROUTINE RDTL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....DETAIL RECORDS.
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
C
      STORE I1 IN DTLV1
      STORE I2 IN DTLV2
      STORE I3 IN DTLV3
      STORE I4 IN DTLV4
      STORE I5 IN DTLV5
      STORE I6 IN DTLV6
      STORE I7 IN DTLV7
      STORE I8 IN DTLV8
      STORE I9 IN DTLV9
      STORE I10 IN DTLV0
C
      LET MORE = MORE - 1
C
      RETURN
      END
```

```
*IBFTC R3
              SUBROUTINE R3
C
C
C.....PURPOSE - TO END RUN.
C
C
C.....IDSUB = 3.
C
      CALL RPT10
      CALL RPT20
      CALL RPT30
C
      CALL RPTTQ
C
 9999 RETURN
      END


*IBFTC R10
              SUBROUTINE R10
C
C
C.....PURPOSE - TO INITIALIZE PERSONNEL QUES.
C
C
C.....IDSUB = 10.
C
C
      FIND FIRST, FOR EACH M IN LQ10, WITH (L11(M)) EQ (VA),
     XWHERE IL, IF NONE, GO TO 10
C
      GO TO 20
C
   10 CREATE L10
      STORE VA IN L11(L10)
      FILE L10 IN LQ10
      GO TO 20
C
   20 FIND FIRST, FOR EACH M IN PQUE, WITH (TP10(M)) EQ (VA),
     XAND (SP10(M)) EQ (VB), WHERE IP, IF NONE, GO TO 21
C
      LET QP10(IP) = QP10(IP) + VC
      GO TO 9999
C
   21 CREATE P10
      LET TP10(P10) = VA
      LET SP10(P10) = VB
      LET QP10(P10) = VC
C
      IF (VB) GR (MAX10), LET MAX10 = VB
C
      FILE P10 IN PQUE
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IDFTC RIOP
              SUBROUTINE RIOP
C
C
C.....PURPOSE - TO CALCULATE MAN SHFT AVAIL,
C                              TOTAL MANHRS THIS PERIOD,
C                              TOTAL MANHRS TO DATE.
C
C
C.....CALLED BY RPTIO.
C
C
      IF (IFLAG) NE (0), GO TO 30
C
   10 LET ISHFT = MAXIO / 7
      LET IDAY  = 1
      LET IFLAG = 1
C
   20 DO TO 200, FOR EACH M IN PQUE, WITH (PFLG(M)) EQ (0)
      IF (SPIO(M)) GR (ISHFT), GO TO 200
      LET DPIO(M) = IDAY
      LET PFLG(M) = 1
  200 REPEAT 20
C
      LET ISHFT = ISHFT + (MAXIO / 7)
      IF (ISHFT) GR (MAXIO), GO TO 30
      LET IDAY = IDAY + 1
      GO TO 20
C
   30 IF (TIO) EQ (7.0), GO TO 37
      IF (TIO) EQ (1.0), GO TO 31
      GO TO 9999
C
   31 LET LOOP = LOOP + 1
      IF (LOOP) EQ (8), GO TO 310
      GO TO 311
C
  310 LET LOOP  = 1
      LET ILOOP = ILOOP + 7
C
  311 LET JDAY = DPART(RTIO) - ILOOP
C
  312 DO TO 319, FOR EACH J IN PQUE, WITH (DPIO(J)) EQ (JDAY)
      FIND FIRST, FOR EACH K IN LOIO, WITH (L11(K)) EQ (TPIO(J)),
     XWHERE IL, IF NONE, GO TO 9999
      LET L12(IL) = L12(IL) + OPIO(J)
      LET V3 = OPIO(J)
      LET L13(IL) = L13(IL) + V3 * SHIFT
      LET L14(IL) = L14(IL) + V3 * SHIFT
  319 REPEAT 312
      GO TO 9999
C
   37 DO TO 370, FOR EACH N IN PQUE
```

```
      FIND FIRST, FOR EACH K IN LQ10, WITH (L11(K)) EQ (TP10(N)),
     XWHERE IL, IF NONE, GO TO 9999
      LET L12(IL) = L12(IL) + QP10(N)
      LET V3 = QP10(N)
      LET L13(IL) = L13(IL) + V3 * SHIFT
      LET L14(IL) = L14(IL) + V3 * SHIFT
  370 REPEAT 37
C
  371 DO TO 374, FOR EACH M IN PQ12
C
  372 DO TO 373, FOR EACH I IN PQUE, WITH (TP10(I)) EQ (TP12(M)),
     XAND (SP10(I)) EQ (SP12(M))
C
      LET QP10(I) = QP10(I) + QP12(M)
C
  373 REPEAT 372
C
      REMOVE M FROM PQ12
      DESTROY P12 CALLED M
C
  374 REPEAT 371
C
      GO TO 9999
C
 9999 RETURN
      END


*IBFTC R20
            SUBROUTINE R20
C
C
C.....PURPOSE - TO INITIALIZE EQUIPMENT QUES.
C
C
C.....IDSUB = 20.
C
C
      FIND FIRST, FOR EACH M IN LQ20, WITH (L21(M)) EQ (VA),
     XWHERE IL, IF NONE, GO TO 10
C
      GO TO 20
C
   10 CREATE L20
      STORE VA IN L21(L20)
      LET L22(L20) = L22(L20) + VC
      LET L23(L20) = L23(L20) + VC
      FILE L20 IN LQ20
      GO TO 9999
C
   20 LET L22(IL) = L22(IL) + VC
      LET L23(IL) = L23(IL) + VC
      GO TO 9999
C
 9999 RETURN
```

```
*IBFTC R30
            SUBROUTINE R30
C
C
C.....PURPOSE - TO INITIALIZE SPARE PARTS QUES.
C
C
C.....IDSUB = 30.
C
C
      FIND FIRST, FOR EACH M IN LQ30, WITH (L31(M)) EQ (VA),
     XWHERE IL, IF NONE, GO TO 10
C
      GO TO 20
C
   10 CREATE L30
      STORE VA IN L31(L30)
      LET L32(L30) = L32(L30) + VC
      LET L33(L30) = L33(L30) + VC
      FILE L30 IN LQ30
      GO TO 9999
C
   20 LET L32(IL) = L32(IL) + VC
      LET L33(IL) = L33(IL) + VC
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R200
              SUBROUTINE R200
C
C
C.....PURPOSE - MAINTENANCE COMPLETED.
C
C
C.....IDSUB = 200.
C
C
      IF (MURG) EQ (0), GO TO 9999
      CALL RDIL
C
    1 DO TO 4, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
C
      LET T200(M) = ETIME
C
    2 DO TO 3, FOR EACH N IN RQUE(M), WITH (RTID(N)) EQ (TRSM),
     XAND (RSUB(N)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN LQ10, WITH (L11(L)) EQ (RTYP(N)),
     XWHERE IL, IF NONE, GO TO 3
C
      LET FQTY = RQTY(N)
      LET L15(IL) = L15(IL) + ((T200(M) - T600(M)) * FQTY)
C
    3 REPEAT 2
C
    4 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R500
            SUBROUTINE R500
C
C
C.....PURPOSE - TEAM DISPATCHED BY BASE CONTROL.
C
C
C.....IDSUB = 500.
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDIL
      IF (DTLV4) EQ (MSITE), GO TO 300
C
C..........SEARCH FOR NOR (NORS/NORE).
C
  100 DO TO 199, FOR EACH M IN NQUE, WITH (NRID(M)) EQ (VC)
C
      IF (NSUB(M)) EQ (1010), GO TO 101
      IF (NSUB(M)) EQ (1020), GO TO 102
      GO TO 198
C
  101 FIND FIRST, FOR EACH L IN L030, WITH (L31(L)) EQ (NTYP(M)),
     XWHERE IL, IF NONE, GO TO 198
C
      LET L36(IL) = L36(IL) + DTLV9 - TNOR(M)
C
      GO TO 198
C
  102 FIND FIRST, FOR EACH L IN L020, WITH (L21(L)) EQ (NTYP(M)),
     XWHERE IL, IF NONE, GO TO 198
C
      LET L26(IL) = L26(IL) + DTLV9 - TNOR(M)
C
      GO TO 198
C
  198 REMOVE M FROM NQUE
      DESTROY NOR CALLED M
C
  199 REPEAT 100
C
C..........CREATE TEAMS ALONG WITH ITS RESOURCES.
C
      FIND FIRST, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM),
     XWHERE IT, IF NONE, GO TO 200
C
      LET TTID(IT) = TRSM
      LET TRID(IT) = VC
      LET T500(IT) = DTLV9
C
      GO TO 201
C
  200 CREATE TEAM CALLED T
```

```
C
      LET TTID(T) = TRSM
      LET TRID(T) = VC
      LET T500(T) = DTLV9
      LET S500(T) = DTLV9
C
      FILE T IN TQUE
C
      GO TO 201
C
  201 DO TO 202, FOR EACH M IN PQUE, WITH (RTID(M)) EQ (TRSM)
C
      FILE M IN RQUE(T)
      REMOVE M FROM BQUE
C
  202 REPEAT 201
C
      GO TO 9999
C
C..........STORE DEPARTURE TIME FROM SITE(T504).
C
  300 DO TO 301, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
C
      LET T504(M) = ETIME
C
  301 REPEAT 300
C
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R600
            SUBROUTINE R600
C
C
C.....PURPOSE - TEAM ARRIVAL AT SITE.
C
C
C.....IDSUB = 600.
C
C
    1 DO TO 4, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
C
      LET T600(M) = ETIME
C
    2 DO TO 3, FOR EACH N IN RQUE(M), WITH (RTID(N)) EQ (TRSM),
     XAND (RSUB(N)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN LQ10, WITH (L11(L)) EQ (RTYP(N)),
     XWHERE IL, IF NONE, GO TO 3
C
      LET FQTY = RQTY(N)
      LET L17(IL) = L17(IL) + ((T600(M) - T500(M)) * FQTY)
C
    3 REPEAT 2
C
    4 REPEAT 1
C
 9999 RETURN
      END
```

```
*IPFTC R700
              SUBROUTINE R700
C
C
C.....PURPOSE - TEAM ARRIVAL AT BASE.
C
C
C.....IDSUP = 700.
C
C
    1 DO TO 4, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
C
      LET T700(M) = ETIME
C
    2 DO TO 3, FOR EACH N IN RQUE(M), WITH (RTID(N)) EQ (TRSM),
     XAND (RSUB(N)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN LQ10, WITH (L11(L)) EQ (RTYP(N)),
     XWHERE IL, IF NONE, GO TO 3
C
      LET FQTY = RQTY(N)
      LET L17(IL) = L17(IL) + ((T700(M) - T504(M)) * FQTY)
C
    3 REPEAT 2
C
    4 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R800
            SUBROUTINE R800
C
C
C.....PURPOSE - TEAM LOST ENROUTE TO SITE/BASE.
C
C
C.....IDSUP = 800
C
C
    1 DO TO 9, FOR EACH IT IN TQUE, WITH (TTID(IT)) EQ (TRSM)
C
      LET T800(IT) = TIME
C
    2 DO TO 8, FOR EACH IR IN RQUE(IT), WITH (RTID(IR)) EQ (TRSM),
     XAND (RSUB(IR)) EQ (1002)
C
      FIND FIRST, FOR EACH ML IN LC10, WITH (L11(ML)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 7
C
      IF (T500(IT)) EQ (0.0), GO TO 5
      IF (T504(IT)) EQ (0.0), GO TO 3
      GO TO 4
C
    3 LET FQTY = RQTY(IR)
      LET L17(IL) = L17(IL) + ((T800(IT) - T500(IT)) * FQTY)
      GO TO 5
C
    4 LET FQTY = RQTY(IR)
      LET L17(IL) = L17(IL) + ((T800(IT) - T504(IT)) * FQTY)
      GO TO 5
C
    5 DO TO 6, FOR EACH IP IN PQUE, WITH (TP10(IP)) EQ (RTYP(IR)),
     XAND (SP10(IP)) EQ (RSFT(IR))
C
      LET QP10(IP) = QP10(IP) - RQTY(IR)
C
    6 REPEAT 5
C
      CALL OVTIME(IT,IR,IL)
C
    7 REMOVE IR FROM RQUE(IT)
      DESTROY RESRC CALLED IR
C
    8 REPEAT 2
C
      IF RQUE(IT) IS NOT EMPTY, GO TO 9
C
      REMOVE IT FROM TQUE
      DESTROY TEAM CALLED IT
C
    9 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1002
              SUBROUTINE R1002
C
C
C.....PURPOSE - PERSONNEL RESOURCES ASSIGNED TO TEAM.
C
C
C.....IDSUB = 1002.
C
C
      CREATE RESRC CALLED R
C
      LET RTID(R) = VA
      LET RRID(R) = TRSM
      LET RTYP(R) = VB
      LET RUTY(R) = VC
      LET RSFT(R) = SXOW
      LET RSUB(R) = IDSUB
C
      FILE R IN BQUE
C
      RETURN
      END



*IBFTC R1010
              SUBROUTINE R1010
C
C
C.....PURPOSE - PARTS STOCKOUT (NORS).
C
C
C.....IDSUB = 1010.
C
C
    1 DO TO 2, FOR EACH M IN LU30, WITH (L31(M)) EQ (VA)
      LET L38(M) = L38(M) + 1
      LET L39(M) = L39(M) + 1
    2 REPEAT 1
C
      CREATE NOR CALLED S
C
      LET NTID(S) = VA
      LET NRID(S) = TRSM
      LET NTYP(S) = VB
      LET NUTY(S) = VC
      LET NSUB(S) = IDSUB
      LET TNOR(S) = ETIME
C
      FILE S IN NQUE
C
 9999 RETURN
      END
```

```
*IBFTC R1012
            SUBROUTINE R1012
C
C
C.....PURPOSE - PARTS ASSIGNED TO TEAM.
C
C
C.....IDSUB = 1012.
C
C
    1 DO TO 2, FOR EACH M IN L030, WITH (L31(M)) EQ (VB)
      LET L33(M) = L33(M) - VC
      LET L34(M) = L34(M) + 1
      LET L35(M) = L35(M) + 1
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1020
            SUBROUTINE R1020
C
C
C.....PURPOSE - EQUIPMENT STOCKOUT (NORE).
C
C
C.....IDSUB = 1020.
C
C
    1 DO TO 2, FOR EACH M IN L020, WITH (L21(M)) EQ (VB)
      LET L28(M) = L28(M) + 1
      LET L29(M) = L29(M) + 1
    2 REPEAT 1
C
      CREATE NOR CALLED E
C
      LET NTID(E) = VA
      LET NRID(E) = TRSM
      LET NTYP(E) = VB
      LET NQTY(E) = VC
      LET NSUB(E) = IDSUB
      LET TNOR(E) = ETIME
C
      FILE E IN NQUE
C
 9999 RETURN
      END
```

```
*IBFTC R1022
               SUBROUTINE R1022
C
C
C.....PURPOSE - EQUIPMENT ASSIGNED TO TEAM.
C
C
C.....IDSUB = 1022.
C
C
    1 DO TO 2, FOR EACH M IN LO20, WITH (L21(M)) EQ (VB)
      LET L23(M) = L23(M) - VC
      LET L24(M) = L24(M) + 1
      LET L25(M) = L25(M) + 1
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1100
               SUBROUTINE R1100
C
C
C.....PURPOSE - EXTRA PARTS ASSIGNED TO TEAM.
C
C
C.....IDSUB = 1100.
C
C
    1 DO TO 2, FOR EACH M IN LO30, WITH (L31(M)) EQ (VP)
      LET L33(M) = L33(M) - VC
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IDFTC R1200
              SUBROUTINE R1200
C
C
C.....PURPOSE - EXOG. PERSONNEL ARRIVAL TO POOL.
C
C
C.....IDSUB = 1200.
C
C
      IF (T10) EQ (1.0), GO TO 1
      IF (T10) EQ (7.0), GO TO 7
      GO TO 9999
C
    1 FIND FIRST, FOR EACH I IN POOF, WITH (TP10(I)) EQ (VB),
     XAND (SP10(I)) EQ (VA), WHERE IP, IF NONE, GO TO 9999
C
      LET QP10(IP) = QP10(IP) + VC
      GO TO 9999
C
    7 LET ISHFT = MAX10 / 7
      LET IDAY  = 1
   71 IF (VA) GR (ISHFT), GO TO 72
      LET JDAY = IDAY
      GO TO 73
C
   72 LET ISHFT = ISHFT + (MAX10 / 7)
      IF (ISHFT) GR (MAX10), GO TO 73
      LET IDAY = IDAY + 1
      GO TO 71
C
   73 LET IDAY = DPART(ETIME)
      IF (IDAY) LE (JDAY), GO TO 1
C
      CREATE P12 CALLED P
C
      STORE VB    IN TP12(P)
      STORE VA    IN SP12(P)
      STORE VC    IN QP12(P)
      STORE JDAY  IN DP12(P)
C
      FILE P IN P012
C
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R1210
              SUBROUTINE R1210
C
C
C.....PURPOSE - EXOG. PARTS ARRIVAL TO POOL.
C
C
C.....IDSUB = 1210.
C
C
    1 DO TO 2, FOR EACH M IN LO30, WITH (L31(M)) EQ (VB)
      LET L32(M) = L32(M) + VC
      LET L33(M) = L33(M) + VC
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1220
              SUBROUTINE R1220
C
C
C.....PURPOSE - EXOG. EQUIPMENT ARRIVAL AT POOL.
C
C
C.....IDSUB = 1220.
C
C
    1 DO TO 2, FOR EACH M IN LO20, WITH (L21(M)) EQ (VB)
      LET L22(M) = L22(M) + VC
      LET L23(M) = L23(M) + VC
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1400
              SUBROUTINE R1400
C
C
C.....PURPOSE - PARTS RETURNED TO POOL.
C
C
C.....IDSUB = 1400.
C
C
    1 DO TO 2, FOR EACH M IN LO30, WITH (L31(M)) EQ (VB)
      LET L33(M) = L33(M) + VC
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1401
              SUBROUTINE R1401
C
C
C.....PURPOSE - PARTS RETURNED TO POOL (REPAIRED).
C
C
C.....IDSUB = 1401.
C
C
    1 DO TO 2, FOR EACH M IN L030, WITH (L31(M)) EQ (VB)
      LET L33(M) = L33(M) + VC
    2 REPEAT 1
C
 9999 RETURN
      END


*IBFTC R1450
              SUBROUTINE R1450
C
C
C.....PURPOSE - PERSONNEL RETURNED TO BASE POOL.
C
C
C.....IDSUB = 1450.
C
C
    1 DO TO 5, FOR EACH IT IN TQUE, WITH (TTID(IT)) EQ (TRSM)
C
      LET T1450(IT) = ETIME
C
    2 DO TO 4, FOR EACH IR IN RQUE(IT), WITH (RTID(IR)) EQ (TRSM),
     XAND (RSUB(IR)) EQ (1002)
C
      FIND FIRST, FOR EACH ML IN LQ10, WITH (L11(ML)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 3
C
      CALL OVTIME(IT,IR,IL)
C
    3 REMOVE IR FROM RQUE(IT)
      DESTROY RESRC CALLED IR
C
    4 REPEAT 2
C
      IF RQUE(IT) IS NOT EMPTY, GO TO 5
C
      REMOVE IT FROM TQUE
      DESTROY TEAM CALLED IT
C
    5 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC R1470
              SUBROUTINE R1470
C
C
C.....PURPOSE - EQUIPMENT RETURNED TO POOL.
C
C
C.....IDSUB = 1470.
C
C
    1 DO TO 2, FOR EACH M IN L220, WITH (L21(M)) EQ (VB)
      LET L23(M) = L23(M) + VC
    2 REPEAT 1
C
 9999 RETURN
      END



*IBFTC R1900
              SUBROUTINE R1900
C
C
C.....PURPOSE - TO SET-UP MSITE VARIABLE TO DETERMINE WHICH
C                DIRECTION TEAM IS TRAVELING (IDSUB-500).
C
C
C.....IDSUB = 1900.
C
C
      IF (VC) GE (MSITE), LET MSITE = VC + 1
C
      RETURN
      END
```

```
*IBFTC OVTIME
              SUBROUTINE OVTIME(IT,IR,IL)
C
C
C.....PURPOSE - TO COMPUTE OVERTIME HOURS.
C
C
C.....IDSUB = 1450/800.
C
C
      IF (T1450(IT)) NE (0.0), GO TO 1
      IF  (T800(IT)) NE (0.0), GO TO 2
C
      GO TO 9999
C
    1 LET OVHRS = T1450(IT)
      GO TO 3
C
    2 LET OVHRS = T800(IT)
      GO TO 3
C
    3 LET IDAY = DPART(S500(IT))
      LET FDAY = IDAY
      LET HSFT = S500(IT) - FDAY
C
      IF (HSFT) LS (0.33333), GO TO 10
      IF (HSFT) LS (0.66666), GO TO 20
      IF (HSFT) LE (1.00000), GO TO 30
C
      GO TO 9999
C
   10 LET HSFT = 0.33333 + FDAY
      GO TO 100
C
   20 LET HSFT = 0.66666 + FDAY
      GO TO 100
C
   30 LET HSFT = 1.00000 + FDAY
      GO TO 100
C
  100 IF (HSFT) GE (OVHRS), GO TO 9999
C
      LET FQTY = RQTY(IR)
      LET L18(IL) = L18(IL) + ((OVHRS - HSFT) * FQTY)
C
 9999 RETURN
      END
```

```
*IBFTC RPT10
            SUBROUTINE RPT10
C
C
C.....PURPOSE - TO REPORT MANHOUR ACCOUNTING.
C
C
C.....CALLED BY RLBL/R3.
C
C
      CALL RTOVT
      CALL RTFLM
      CALL RTINT
C
      CALL HDG10
C
      CALL RIOP
C
    1 DO TO 2, FOR EACH L IN LO10
      LET L15(L) = L15(L) * 24.0
      LET L17(L) = L17(L) * 24.0
      LET L18(L) = L18(L) * 24.0
      LET L15A(L) = L15A(L) + L15(L)
      LET L17A(L) = L17A(L) + L17(L)
      LET L18A(L) = L18A(L) + L18(L)
      LET L19(L) = (L15(L) + L17(L) + L18(L)) / L13(L)
      LET L110(L) = (L15A(L) + L17A(L) + L18A(L)) / L14(L)
C
      CALL RPG10(L)
C
      LET L12(L) = 0
      LET L13(L) = 0.0
      LET L15(L) = 0.0
      LET L17(L) = 0.0
      LET L18(L) = 0.0
C
    2 REPEAT 1
C
      LET RT10 = RT10 + T10
C
 9999 RETURN
      END
```

```
*IBFTC RTFLM
           SUBROUTINE RTFLM
C
C
C.....PURPOSE - TO COMPUTE FLT LINE MAINT HRS AS OF REPORTING TIME.
C
C
C.....CALLED BY RPT10.
C
C
    1 DO TO 5, FOR EACH IT IN TQUE, WITH (T600(IT)) NE (0.0),
     X                                AND (T200(IT)) EQ (0.0)
C
    2 DO TO 3, FOR EACH IR IN RQUE(IT), WITH (RSUB(IR)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN L010, WITH (L11(L)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 4
C
      LET FQTY = RQTY(IR)
      LET L15(IL) = L15(IL) + ((RT10 - T600(IT)) * FQTY)
C
    3 REPEAT 2
C
    4 LET T600(IT) = RT10
C
    5 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC RTINT
            SUBROUTINE RTINT
C
C
C.....PURPOSE - TO COMPUTE INTRAN HOURS AS OF REPORTING TIME.
C
C
C.....CALLED BY RPT10.
C
C
    1 DO TO 5, FOR EACH IT IN TQUE, WITH (T500(IT)) NE (0.0),
     X                                   AND (T600(IT)) EQ (0.0)
C
    2 DO TO 3, FOR EACH IR IN RQUE(IT), WITH (RSUB(IR)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN LO10, WITH (L11(L)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 4
C
      LET FQTY = RQTY(IR)
      LET L17(IL) = L17(IL) + ((RT10 - T500(IT)) * FQTY)
C
    3 REPEAT 2
C
    4 LET T500(IT) = RT10
C
    5 REPEAT 1
C
   11 DO TO 15, FOR EACH IT IN TQUE, WITH (T504(IT)) NE (0.0),
     X                                   AND (T700(IT)) EQ (0.0)
C
   12 DO TO 13, FOR EACH IR IN RQUE(IT), WITH (RSUB(IR)) EQ (1002)
C
      FIND FIRST, FOR EACH L IN LO10, WITH (L11(L)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 14
C
      LET FQTY = RQTY(IR)
      LET L17(IL) = L17(IL) + ((RT10 - T504(IT)) * FQTY)
C
   13 REPEAT 12
C
   14 LET T504(IT) = RT10
      LET T500(IT) = RT10
C
   15 REPEAT 11
C
 9999 RETURN
      END
```

```
*IBFTC RTOVT
              SUBROUTINE RTOVT
C
C
C.....PURPOSE - TO COMPUTE OVERTIME HOURS AS OF REPORTING TIME.
C
C
C.....CALLED BY RPT10.
C
C
      LET OVHRS = RT10
C
    1 DO TO 2000, FOR EACH IT IN TOUF, WITH (S500(IT)) NE (0.0)
C
      LET IDAY = DPART(S500(IT))
      LET FDAY = IDAY
      LET HSFT = S500(IT) - FDAY
C
      IF (HSFT) LS (0.33333), GO TO 10
      IF (HSFT) LS (0.66666), GO TO 20
      IF (HSFT) LE (1.00000), GO TO 30
C
      GO TO 2000
C
   10 LET HSFT = 0.33333 + FDAY
      GO TO 100
C
   20 LET HSFT = 0.66666 + FDAY
      GO TO 100
C
   30 LET HSFT = 1.00000 + FDAY
      GO TO 100
C
  100 IF (HSFT) GE (OVHRS), GO TO 2000
C
  200 DO TO 1000, FOR EACH IR IN ROUF(IT), WITH (RSUB(IR)) EQ (1902)
C
      FIND FIRST, FOR EACH L IN LOLO, WITH (LIIL(L)) EQ (RTYP(IR)),
     XWHERE IL, IF NONE, GO TO 2000
C
      LET FQTY = RQTY(IR)
      LET LIB(IL) = LIB(IL) + ((OVHRS - HSFT) * FQTY)
C
 1000 REPEAT 200
C
 2000 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC RPT20
              SUBROUTINE RPT20
C
C
C.....PURPOSE - TO REPORT MAINT. EQUIPMENT USAGE.
C
C
C.....CALLED BY RLBL/R3.
C
C
      CALL RTNORE
C
      CALL HDG20
C
    1 DO TO 2, FOR EACH L IN LQ20
      LET L27(L) = L27(L) + L26(L)
C
      CALL RPG20(L)
C
      LET L24(L) = 0
      LET L26(L) = 0.0
      LET L28(L) = 0
C
    2 REPEAT 1
C
      LET RT20 = RT20 + T20
C
 9999 RETURN
      END



*IBFTC RTNORE
              SUBROUTINE RTNORE
C
C
C.....PURPOSE - TO COMPUTE NORE TIME AS OF REPORTING TIME.
C
C
C.....CALLED BY RPT20.
C
C
    1 DO TO 2, FOR EACH N IN NQUE, WITH (NSUB(N)) EQ (1020)
C
      FIND FIRST, FOR EACH L IN LQ20, WITH (L21(L)) EQ (NTYP(N)),
     XWHERE IL, IF NONE, GO TO 2
C
      LET L26(IL) = L26(IL) + RT  ) - TNOR(N)
      LET TNOR(N) = RT20
C
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC RPT30
                SUBROUTINE RPT30
C
C
C.....PURPOSE - TO REPORT SPARE PARTS USAGE.
C
C
C.....CALLED BY RLBL/R3.
C
C
      CALL RTNORS
C
      CALL HDG30
C
    1 DO TO 2, FOR EACH L IN LQ30
      LET L37(L) = L37(L) + L36(L)
C
      CALL RPG30(L)
C
      LET L34(L) = 0
      LET L36(L) = 0.0
      LET L38(L) = 0
C
    2 REPEAT 1
C
      LET RT30 = RT30 + T30
C
 9999 RETURN
      END



*IBFTC RTNORS
                SUBROUTINE RTNORS
C
C
C.....PURPOSE - TO COMPUTE NORS TIME AS OF REPORTING TIME.
C
C
C.....CALLED BY RPT30.
C
C
    1 DO TO 2, FOR EACH N IN NQUE, WITH (NSUB(N)) EQ (1010)
C
      FIND FIRST, FOR EACH L IN LQ30, WITH (L31(L)) EQ (NTYP(N)),
     XWHERE IL, IF NONE, GO TO 2
C
      LET L36(IL) = L36(IL) + RT30 - TNOR(N)
      LET TNOR(N) = RT30
C
    2 REPEAT 1
C
 9999 RETURN
      END
```

```
*IBFTC RPTTQ
      SUBROUTINE RPTTQ
C
C
C.....PURPOSE - TO PRINT ALL MEMBERS OF TQUES LEFT AT END-SIM
C
C
C.....CALLED BY R3.
C
C
      CALL HDGTQ
C
    1 DO TO 2, FOR EACH I IN TQUE
      CALL RPGTQ(I)
    2 REPEAT 1
C
      RETURN
      END


*IBFTC HDG10
      REPORT HDG10
*                                                         MANHOUR ACCOUNTING FOR
 *
*              MAN SHFT     TOTAL MANHRS        TOTAL MANHRS          FLT  LINE
*     TYPE       AVAIL      THIS PERIOD           TO DATE             MAINT HRS
      END
                                                                              *
 PERIOD ENDING ****.**                                                    2
                       RT10
      INTRAN       OVERTIME      UTIL FACTOR        UTIL FACTOR
       HOURS          HOURS      THIS PERIOD          TO DATE              1
      END



*IBFTC RPG10
      REPORT RPG10(L)
*        *        *            *.**              *.**                 *.**
 *    L11(L),L12(L),L13(L),L14(L),L15(L),L17(L),L18(L),L19(L),L110(L)
      END
                                                                              *
        *.**          *.**          *.**              *.**              *
                                                                              *
      END
```

```
*IBFTC HDG20
      REPORT HDG20
*                                                        MAINT. EQUIPMENT USAGE FOR
 *
*                  AUTH.     TOTAL QTY        SITE DEMANDS      SITE DEMANDS
*      TYPE        QTY       AVAILABLE        THIS  PERIOD       TO  DATE
       END
                                                                                 *
PERIOD ENDING ****.**                                                          2
                  RT20
   NORE  TIME        NORE TIME        NO. NORE      NO. NORE
   THIS PERIOD        TO DATE         THIS PER      TO  DATE                   1
       END


*IBFTC RPG20
      REPORT RPG20(L)
*       *       *             *               *                 *
 *    L21(L),L22(L),L23(L),L24(L),L25(L),L26(L),L27(L),L28(L),L29(L)
      END
                                                                                 *
           *.**          *.**           *           *             *
                                                                                 *
      END



*IBFTC HDG30
      REPORT HDG30
*                                                            SPARE PARTS USAGE FOR
 *
*                  AUTH.     TOTAL QTY        SITE DEMANDS      SITE DEMANDS
*      TYPE        QTY       AVAILABLE        THIS  PERIOD       TO  DATE
       END
                                                                                 *
PERIOD ENDING ****.**                                                          2
                  RT30
   NORS  TIME        NORS TIME        NO. NORS      NO. NORS
   THIS PERIOD        TO DATE         THIS PER      TO  DATE                   1
       END
```

```
*IBFTC RPG30
      REPORT RPG30(L)
*        *      *             *                  *                 *
 *    L31(L),L32(L),L33(L),L34(L),L35(L),L36(L),L37(L),L38(L),L39(L)
      END
                                                                    *
         *.**        *.**           *            *            *
                                                                    *
      END



*IBFTC HDGTQ
      REPORT HDGTQ
*     MEMBERS OF TQUES
*       TID   RID        T500        T600        T200        T504        T700
      END
                                                                    *
                                                                 2
        T800      T1450                                          2
      END



*IBFTC RPGTQ
      REPORT RPGTQ(I)
*           *    *    *.*****    *.*****    *.*****    *.*****    *.*****
 *    TTID(I),TRID(I),T500(I),T600(I),T200(I),T504(I),T700(I),T800(I),T1
      END
                                                                    *
    *.*****     *.*****                                            *
450(I)
      END
```

INITIALIZATION DECK

```
*ENTRY            MAIN
1           50
   1  22      Z
  23          R
  24          R                                                       7.00000
  25          R                                                       7.00000
  26  28      Z                                                       7.00000
  29          R
  30  32      Z                                                       8.00000
  33          R
  34        1 R   100   33                                          100
00                                                                      (12.
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
```

95
96
97
98
99
35 50 7

Programs 4 to 7

AIRCRAFT RECOVERY PACKAGE

The "recovery package" deals with the problem of maintenance quality. The displays that follow refer to the measures associated with the recovery of an aircraft from the effects of the previous sortie.

The best way to determine maintenance quality is to determine its impact on the stated mission of the weapon system. Good maintenance results in high mission capability, poor maintenance degrades mission capability. Most measures of maintenance quality contain this idea by implication, generally being stated as an index of capability to deliver clean aircraft quickly.

The sortie is also a major factor in aircraft maintenance. The sortie, more than any other factor, determines maintenance actions. Consequently, maintenance quality is more intimately associated with the aircraft's recovery.

Analyzing sortie data is a two-step process. The first step is to process the label record tape data by use of a "recovery program." This results in the generation of an "Aircraft Output Tape," (format described in Fig. 14), which is used to generate the selected Program(s) 4-6. This tape must be sorted on tail number and "time job started."

INITIALIZATION

The variables description and initialization table, which follows, contains the information required to initialize the program. An example initialization data deck listing follows the "OUTPUT PROGRAM" section. The program requires initialization of 37 variables; only 5 require values (see Table 4). Array 23 specifies the quantity of bases to be analyzed. Array 24 lists the base numbers. Array 26 specifies the total quantity of failure levels to be included in the report. Array 27 specifies the failure level numbers. Array 29 specifies the time that the report is to terminate. All of the other Arrays are set to zero.

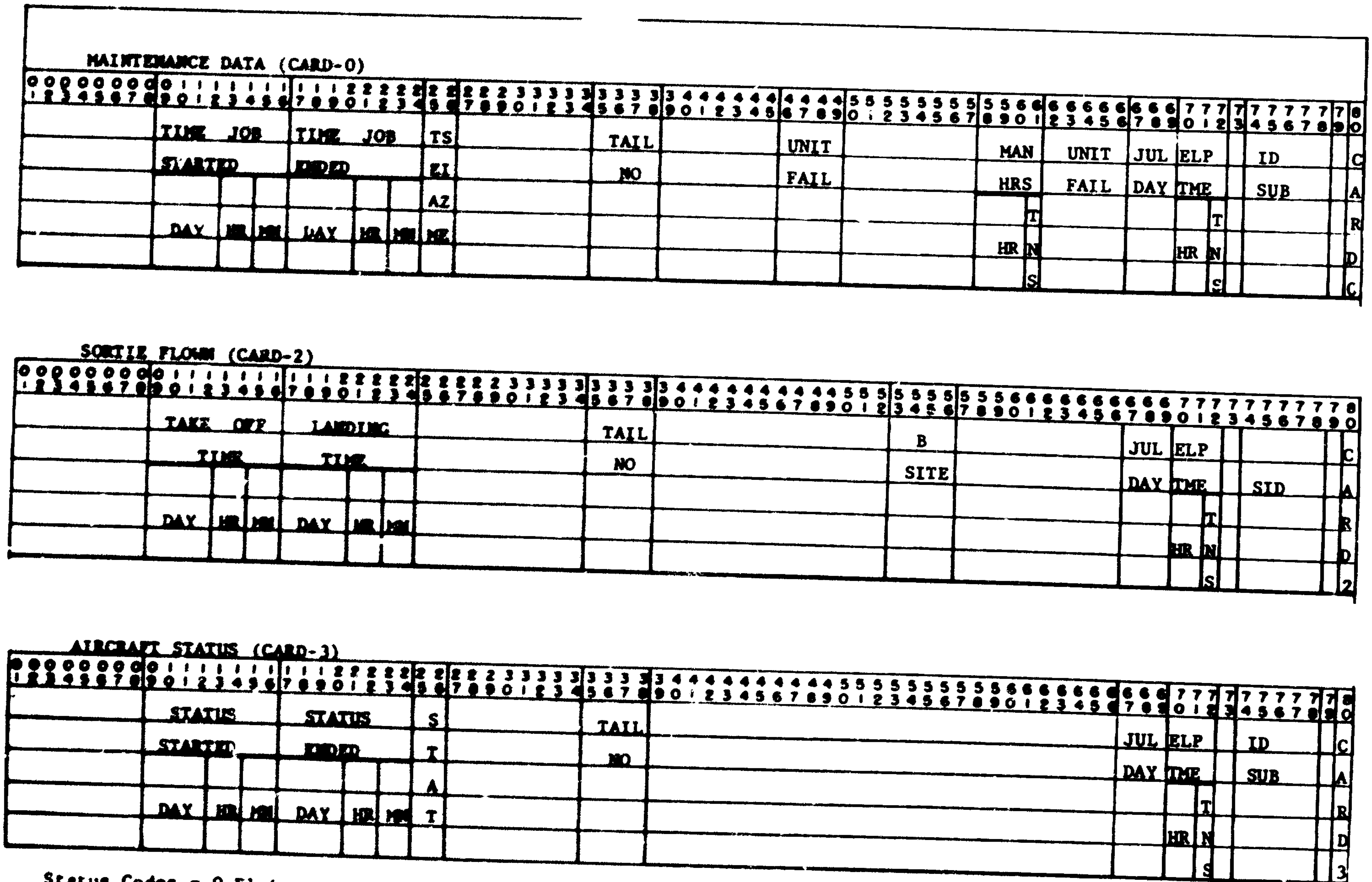
OUTPUT PROGRAM DESCRIPTION

The input to this program is the tape generated by the ABC Simulator.

Table 4

VARIABLE DESCRIPTION AND INITIALIZATION:
AIRCRAFT RECOVERY SORT

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col.: 19-22 (rows)	Array Number of Attribute to Be Entered in Fig. 5 Col.: 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1-22	0			X										
23	0	I			V						Number of bases to be analyzed.	BASES	E	
24	1	I			V			23			Specify each base code to be analyzed.	BASE		A
25	0			X										
26	0	I			V						Number of failure levels to be analyzed.	FLVLS	E	
27	1	I			V			26			Specify each failure level to be analyzed	FLVL		A
28	0			X										
29	0				V						Time to end report other than at END-SIM	[illegible]	E	
30-31	0			X										



Status Codes = 0 Flying
1 Off-Alert
2 On-Alert

Fig. 14 -- Format for aircraft output tape

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may be run at one time.

When a label record is read, the value of the failure level is compared to the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

If ETIME is greater than TMEND, the program is then terminated.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-PR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/wk
SXDW = S/wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRSM = ID Addresses
MORE = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-PR).

DTLV1 = Integer variable 1
DTLV2 = Integer variable 2
DTLV3 = Integer variable 3
DTLV4 = Integer variable 4
DTLV5 = Integer variable 5
DTLV6 = Integer variable 6
DTLV7 = Integer variable 7
DTLV8 = Integer variable 8
DTLV9 = Float variable 1
DTLV0 = Float variable 2

Base table.

BASES = Number of base codes to be processed.

BASE = Base codes to be processed.
BFLAG = Controls flow of events as a result of EBAS vs BASES.

Failure level table.

FLVLS = Number of failure codes to be processed.
FLVL = Failure level codes to be processed.
TMEND = Time initialized to end this run prematurely.

SETS

Name = MQUE used for maintenance events. No subscript. Ranked on ETIME.

Owner = SIMSCRIPT system.

Member = MAINT.
M1 = Start time.
M2 = End time.
M3 = Team size.
M4 = Tail no.
M5 = Unit that failed.
M6 = Team ID.
M7 = Request ID.
M8 = Site ID.
M9 = IDSUB.

Name = SQUE used for site events. No subscript. Ranked on SID.

Owner = SIMSCRIPT system.

Member = SITES.
SID = Site ID.
ANO = A Site No.
BNO = B Site No.
SMOD = Mode of site.
SERNO = Tail No.
STATS = Status of site.
STIME = Start time of status.
NFAIL = Number of failures at site.

Name = SRTQ used for sorties. No subscript. Ranked on S1.

Owner = SIMSCRIPT system.

Member = SORTE.
S1 = Start time.
S2 = End time.
S3 = B Site No.
S4 = Tail No.
S8 = Site ID.

Name = TQUE used for team events. No subscript. Ranked on TTID.

Owner = SIMSCRIPT system.

Member = TEAM.
TTID = Team ID.
TQTY = Team size.

```
+                                    1IDSOR 0    I
+                                    2IDSUP 0    I
+                                    4DAYW  0    I
+                                    3SHFT  0    I
+                                    5SXDW  0    I
+                                    6EBAS  0    I
+                                    7VA    0    I
+                                    8VB    0    I
+                                    9VC    0    I
+                                   10TRSM  0    I
+                                   11MORE  0    I
+                                   12LTIME 0    F
+                                   13DTLV1 0    I
+                                   14DTLV2 0    I
+                                   15DTLV3 0    I
+                                   16DTLV4 0    I
+                                   17DTLV5 0    I
+                                   18DTLV6 0    I
+                                   19DTLV7 0    I
+                                   20DTLV8 0    I
+                                   21DTLV9 0    F
+                                   22DTLV0 0    F
+                                   23BASES E    I
+                                   24BASE  1    I
+                                   25BFLAG 0    I
+                                   26FLVLS E    I
+                                   27FLVL  1    I
+                                   28MSITE 0    I
+                                   29TMEND 0    F
+                                   30FSQUE 0    I
+                                   31LSQUE 0    I
+                                   32FMQUE 0    I
+                                   33LMQUE 0    I
+                                   34FSRTQ 0    I
+                                   35LSRTQ 0    I
+                                   36FTQUE 0    I
+                                   37LTQUE 0    I
+                                                         MQUEO   *M1     L
+T MAINT8   8        T M1      1    F
+                    T M2      2    F
+                    T M3      4    I
+                    T M4      5    I
+                    T M5      6    I
+                    T M6      7    I
+                    T M7      8    I
+                    T M8     31    I
+                    T M9     32    I
+                    T SMQUE 33    I
+                    T PMQUE 34    I
+                                                         SQUEO   *SID    L
+T SITES8   8        T SID     1    I
+                    T SNO     2    I
```

```
+                    T BNO    4      I
+                    T SMOD   5      I
+                    T SERNO  6      I
+                    T STATS  7      I
+                    T STIME  8      F
+                    T NFAIL31       I
+                    T PSQUE32       I
+                    T SSQUE33       I
+                                                       SRTGO   *S1     L
+T SORTE8
+                    T S1     1      F
+                    T S2     2      F
+                    T S3     3      I
+                    T S4     4      I
+                    T S8     5      I
+                    T PSRTQ  6      I
+                    T SSRTQ  7      I
+                                                       TQUEO   *TTID   L
+T TEAM 8
+                    T TTID   1      I
+                    T TQTY   2      I
+                    T PTQUE  3      I
+                    T STQUE  4      I
*DEFTC MAIN
              MAIN ROUTINE
C
C
C      .....PLANET - AIRCRAFT RECOVERY TAPE INPUT.....
C
C
C.....PURPOSE  - TO CREATE AIRCRAFT STATUS TAPE
C                 FOR INPUT TO RECOVERY PACKAGE.
C
C
C.....INPUT    - TAPE FROM ABC MODEL.
C
C
C.....OUTPUT   - A/C STATUS TAPE.
C
C
C
      REWIND 8
      REWIND 9
C
    1 CALL RERL
      CALL SELECT
C
      IF (IDSUB) EQ (    3), GO TO 3
      IF (BFLAG) NE (    0), GO TO 9999
      IF (IDSUB) EQ (  110), GO TO 110
      IF (IDSUB) EQ (  200), GO TO 200
      IF (IDSUB) EQ (  500), GO TO 500
```

```
      IF (IDSUB) EQ ( 600), GO TO 600
      IF (IDSUB) EQ ( 800), GO TO 800
      IF (IDSUB) EQ (1002), GO TO 1002
      IF (IDSUB) EQ (1450), GO TO 1450
      IF (IDSUB) EQ (1900), GO TO 1900
      IF (IDSUB) EQ (2000), GO TO 2000
      IF (IDSUB) EQ (2100), GO TO 2100
      IF (IDSUB) EQ (2150), GO TO 2150
      IF (IDSUB) EQ (2300), GO TO 2300
      IF (IDSUB) EQ (2400), GO TO 2400
      IF (IDSUB) EQ (2500), GO TO 2500
      IF (IDSUB) EQ (3100), GO TO 3100
      IF (IDSUB) EQ (3200), GO TO 3200
C
      GO TO 9999
C
    3 CALL R3
      CALL EXIT
C
  110 CALL R110
      GO TO 9999
C
  200 CALL R200
      GO TO 9999
C
  500 CALL R500
      GO TO 9999
C
  600 CALL R600
      GO TO 9999
C
  800 CALL R800
      GO TO 9999
C
 1002 CALL R1002
      GO TO 9999
C
 1450 CALL R1450
      GO TO 9999
C
 1900 CALL R1900
      GO TO 9999
C
 2000 CALL R2000
      GO TO 9999
C
 2100 CALL R2100
      GO TO 9999
C
 2150 CALL R2150
      GO TO 9999
C
```

```
 2300 CALL R2300
      GO TO 9999
C
 2400 CALL R2400
      GO TO 9999
C
 2500 CALL R2500
      GO TO 9999
C
 3100 CALL R3100
      GO TO 9999
C
 3200 CALL R3200
      GO TO 9999
C
 9999 IF (MORE) EQ (0), GO TO 1
      CALL RDTL
      GO TO 9999
C
      END
*IBFTC RLBL
              SUBROUTINE RLBL
C
C.........READS S-PHASE TAPE(9) (BIN MODE).....LABEL RECORDS.
C
      LET BFLAG = 0
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
C
      STORE I1 IN IDSOR
      STORE I2 IN IDSUB
      STORE I3 IN SHFT
      STORE I4 IN DAYW
      STORE I5 IN SXDW
      STORE I6 IN PHAS
      STORE I7 IN VA
      STORE I8 IN VB
      STORE I9 IN VC
      STORE I10 IN TRSM
      STORE I11 IN MORE
      STORE I12 IN ETIME
C
      IF (ETIME) GR (TMEND), GO TO 1
      GO TO 9999
C
    1 CALL R3
      CALL EXIT
C
 9999 RETURN
      END
*IBFTC SELECT
              SUBROUTINE SELECT
```

```
C
C
C.....PURPOSE - TO SELECT EVENTS BY BASE.
C
C
      DO TO 1, FOR EACH BASES I
      IF (EBAS) EQ (BASE(I)), GO TO 2
    1 LOOP
      LET BFLAG = 1
      GO TO 9999
C
    2 LET BFLAG = 0
      GO TO 9999
C
 9999 RETURN
      END
*IBFTC RDTL
             SUBROUTINE RDTL
C
C........READS S-PHASE TAPE(9) (BIN MODE).....DETAIL RECORDS.
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
C
      STORE I1 IN DTLV1
      STORE I2 IN DTLV2
      STORE I3 IN DTLV3
      STORE I4 IN DTLV4
      STORE I5 IN DTLV5
      STORE I6 IN DTLV6
      STORE I7 IN DTLV7
      STORE I8 IN DTLV8
      STORE I9 IN DTLV9
      STORE I10 IN DTLV0
C
      LET MORE = MORE - 1
C
      RETURN
      END
*IBFTC R3
             SUBROUTINE R3
C
C
C.....PURPOSE - TO CLOSE-OUT AND END R-PHASE.
C
C
C.....IDSUB = 3.
C
C
      REWIND TAPE 9
C
      ENDFILE TAPE 8
C
```

```
       REWIND TAPE 8
C
       CALL MOHG
C
     1 GO TO 2, FOR EACH M IN MQUE
C
       LET DTIME = M1(M)
       CALL CLK1(DTIME,I1,I2,I3)
       LET IS1 = I1
       LET IS2 = I2
       LET IS3 = I3
C
       LET DTIME = M2(M)
       CALL CLK1(DTIME,I1,I2,I3)
       LET IE1 = I1
       LET IE2 = I2
       LET IE3 = I3
C
       CALL MQRPG(M,IS1,IS2,IS3,IE1,IE2,IE3)
C
     2 REPEAT 1
C
  9999 RETURN
       END
*IBFTC R110
              SUBROUTINE R110
C
C
C.....PURPOSE - START MAINTENANCE FOR.....
C
C                   (1) EXOG. FAILURE.
C                   (2) EXOG. PM.
C                   (3) EXOG. OVERHAUL.
C
C
C.....IDSUB = 110.
C
C
C
       FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
      XWHERE IS, IF NONE, GO TO 9999
C
       IF (VC) EQ (2), GO TO 2
       IF (VC) EQ (4), GO TO 4
       IF (VC) EQ (6), GO TO 6
C
       GO TO 9999
C
C..........EXOG. FAILURE.
C
     2 DO TO 20, FOR EACH FLVLS I
       IF (VR) EQ (FLVL(I)), GO TO 21
```

```
   20 LOOP
      GO TO 9999
C
   21 IF (NFAIL(IS)) NE (0), GO TO 22
      CALL CRO3(IS)
      LET STATS(IS) = 1
      LET STIME(IS) = ETIME
C
   22 LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE MAINT CALLED E2
C
      STORE SERNO(IS) IN M4(E2)
      STORE ETIME     IN M1(E2)
      STORE VA        IN M5(E2)
      STORE TRSM      IN M8(E2)
      STORE IDSUB     IN M9(E2)
C
      FILE E2 IN MQUE
C
      GO TO 9999
C
C..........EXOG. PM.
C
    4 CREATE MAINT CALLED E4
C
      STORE SERNO(IS) IN M4(E4)
      STORE ETIME     IN M1(E4)
      STORE VA        IN M5(E4)
      STORE TRSM      IN M8(E4)
      STORE IDSUB     IN M9(E4)
C
      FILE E4 IN MQUE
C
      GO TO 9999
C
C..........EXOG. OVERHAUL.
C
    6 CREATE MAINT CALLED E6
C
      STORE SERNO(IS) IN M4(E6)
      STORE ETIME     IN M1(E6)
      STORE VA        IN M5(E6)
      STORE TRSM      IN M8(E6)
      STORE IDSUB     IN M9(E6)
C
      FILE E6 IN MQUE
C
      GO TO 9999
C
 9999 RETURN
      END
```

```
*IBFTC R200
                 SUBROUTINE R200
C
C
C.....PURPOSE - END MAINTENANCE.
C
C
C.....IDSUB = 200.
C
C
C
      IF (MORF) EQ (0), GO TO 9999
      CALL RDTL
C
      FIND FIRST, FOR EACH M IN MQUE, WITH (M6(M)) EQ (TRSM),
     X                                 AND (M8(M)) EQ (VA),
     XWHERE IM, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (DTLV1),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (NFAIL(IS)) NE (1), GO TO 1
C
      CALL CRD3(IS)
      LET STATS(IS) = 2
      LET STIME(IS) = ETIME
C
    1 STORE ETIME IN M2(IM)
C
    2 DO TO 3, FOR EACH I IN TQUE, WITH (TTID(I)) EQ (TRSM)
      LET M3(IM) = M3(IM) + TQTY(I)
    3 REPEAT 2
C
      CALL CRD0(IM)
C
      LET NFAIL(IS) = NFAIL(IS) - 1
C
      REMOVE IM FROM MQUE
      DESTROY MAINT CALLED IM
C
 9999 RETURN
      END
*IBFTC R500
                 SUBROUTINE R500
C
C
C.....PURPOSE - TEAM DISPATCH BY BASE CONTROL.
C
C
C.....IDSUB = 500.
C
C
```

```
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RUTL
C
      IF (DTLV4) EQ (MSITE), GO TO 9999
C
      FIND FIRST, FOR EACH M IN MQUL, WITH (M7(M)) EQ (VC),
     X                                  AND (M8(M)) EQ (DTLV1),
     XWHERE IM, IF NONE, GO TO 9999
C
      STORE TRSM IN M6(IM)
C
 9999 RETURN
      END
*IDFTC R600
             SUBROUTINE R600
C
C
C.....PURPOSE - TEAM ARRIVAL AT SITE.
C
C
C.....IDSUB = 600.
C
C
C
      FIND FIRST, FOR EACH M IN MQUL, WITH (M6(M)) EQ (TRSM),
     X                                  AND (M8(M)) EQ (VA),
     XWHERE IM, IF NONE, GO TO 9999
C
      STORE ETIME IN M1(IM)
C
 9999 RETURN
      END
*IDFTC R800
             SUBROUTINE R800
C
C
C.....PURPOSE - TEAM LOST ENROUTE.
C
C
C.....IDSUB = 800.
C
C
C
    1 DO TO 2, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
      REMOVE M FROM TQUE
      DESTROY TEAM CALLED M
    2 REPEAT 1
C
 9999 RETURN
      END
*IDFTC R1002
```

```
            SUBROUTINE R1002
C
C
C.....PURPOSE - PERSONNEL ASSIGNED TO TEAMS (CREW SIZE).
C
C
C.....IDSUB = 1002.
C
C
C
      FIND FIRST, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (VA),
     XWHERE IT, IF NONE, GO TO 1
C
      LET TQTY(M) = TQTY(M) + VC
      GO TO 9999
C
    1 CREATE TEAM CALLED T
      STORE VA   IN TTID(T)
      STORE VC   IN TQTY(T)
      FILE T IN TQUE
      GO TO 9999
C
 9999 RETURN
      END
*IBFTC R1450
            SUBROUTINE R1450
C
C
C.....PURPOSE - PERSONNEL RETURNED TO BASE POOL.
C
C
C.....IDSUB = 1450.
C
C
C
    1 GO TO 2, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
      REMOVE M FROM TQUE
      DESTROY TEAM CALLED M
    2 REPEAT 1
C
 9999 RETURN
      END
*IBFTC R1900
            SUBROUTINE R1900
C
C
C.....PURPOSE - GENERATE SITES AT BASE.
C
C
C.....IDSUB = 1900.
C
C
```

```
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      CREATE SITES CALLED S
C
      STORE TRSM  IN SID(S)
      STORE VA    IN ANO(S)
      STORE VB    IN BNO(S)
      STORE VC    IN SMOD(S)
      STORE DTLVI IN SERNO(S)
C
      LET STATS(S) = 2
      LET STIME(S) = FTIME
C
      IF (VC) GE (MSITE), LET MSITE = VC + 1
C
      FILE S IN SQUE
C
      LET I999   = 999
      LET I99999 = 99999
      LET ITAILN = DTLVI
      LET ICODE  = 2
C
      WRITE ON TAPE 8, I999, I99999, ITAILN, I999, ICODE
      FORMAT (S8,I3,I5,S18,I4,S28,I3,S10,I1)
C
      GO TO 9999
C
 9999 RETURN
      END
*IBFTC R2000
              SUBROUTINE R2000
C
C
C.....PURPOSE - REQUEST FOR PM.
C
C
C.....IDSUB = 2000.
C
C
C
      FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      CREATE MAINT CALLED P
C
      STORE SERNO(IS) IN M4(P)
      STORE TRSM      IN M8(P)
      STORE FTIME     IN M1(P)
      STORE VB        IN M5(P)
      STORE IDSUB     IN M9(P)
C
```

```
      FILE P  IN MQUE
C
 9999 RETURN
      END
*IBFTC R2100
              SUBROUTINE R2100
C
C
C.....PURPOSE - START MAINT. ON PM.
C
C
C.....IDSUB = 2100.
C
C
C
      FIND FIRST, FOR EACH M IN MQUE, WITH (M6(M)) EQ (VC),
     X                                   AND (M8(M)) EQ (TRSM),
     XAND (M5(M)) EQ (VB), WHERE IM, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (NFAIL(IS)) NE (0), GO TO 1
      CALL CRD3(IS)
      LET STATS(IS) = 1
      LET STIME(IS) = ETIME
C
    1 STORE ETIME IN MI(IM)
      LET NFAIL(IS) = NFAIL(IS) + 1
C
 9999 RETURN
      END
*IBFTC R2150
              SUBROUTINE R2150
C
C
C.....PURPOSE - START MAINT. FOR FAILURE CAUSED BY PM.
C
C
C.....IDSUB = 2150
C
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (NFAIL(IS)) NE (0), GO TO 3
```

```
      CALL CRU3(IS)
      LET STATS(IS) = 1
      LET STIME(IS) = ETIME
C
    3 LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE MAINT CALLED M
C
      STORE SERNO(IS) IN M4(M)
      STORE ETIME     IN M1(M)
      STORE VB        IN M5(M)
      STORE TRSM      IN M8(M)
      STORE IDSUB     IN M9(M)
C
      FILE M IN MQUE
C
 9999 RETURN
      END
*IBFTC R2300
             SUBROUTINE R2300
C
C
C.....PURPOSE - START MAINT. FOR FAILURE CAUSED BY CONTINUOUS MONITOR.
C
C
C.....IDSUB = 2300.
C
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH N IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (NFAIL(IS)) NE (0), GO TO 3
      CALL CRU3(IS)
      LET STATS(IS) = 1
      LET STIME(IS) = ETIME
C
    3 LET NFAIL(IS) = NFAIL(IS) + 1
C
      CREATE MAINT CALLED M
C
      STORE SERNO(IS) IN M4(M)
      STORE ETIME     IN M1(M)
      STORE VB        IN M5(M)
      STORE TRSM      IN M8(M)
      STORE IDSUB     IN M9(M)
C
```

```
      FILE M IN MQUE
C
 9999 RETURN
      END
*IBFTC R2400
            SUBROUTINE R2400
C
C
C.....PURPOSE - RESOURCE REQUEST FOR FAILURES.
C
C
C.....IDSUB = 2400.
C
C
C
      IF (MORE) EQ (0), GO TO 9999
      CALL RDTL
C
      IF (DTLV1) EQ (1), GO TO 1
      IF (DTLV1) EQ (2), GO TO 246
      IF (DTLV1) EQ (3), GO TO 3
      IF (DTLV1) EQ (4), GO TO 246
      IF (DTLV1) EQ (5), GO TO 5
      IF (DTLV1) EQ (6), GO TO 246
      GO TO 9999
C
    1 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2300),
     XAND (M8(M)) EQ (TRSM), AND (M5(M)) EQ (VB), WHERE IM, IF NONE,
     XGO TO 9999
C
      STORE VC IN M7(IM)
      GO TO 9999
C
  246 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (110),
     X                                    AND (M8(M)) EQ (TRSM),
     X                                    AND (M5(M)) EQ (VB),
     XWHERE IM, IF NONE, GO TO 2460
C
      STORE VC IN M7(IM)
      GO TO 9999
C
 2460 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2000),
     X                                    AND (M8(M)) EQ (TRSM),
     X                                    AND (M5(M)) EQ (VB),
     XWHERE IM, IF NONE, GO TO 9999
C
      STORE VC IN M7(IM)
      GO TO 9999
C
    3 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2500),
     XAND (M8(M)) EQ (TRSM), AND (M5(M)) EQ (VB), WHERE IM, IF NONE,
     XGO TO 9999
```

```
C
      STORE VC    IN M7(IM)
      STORE ETIME IN M1(IM)
C
      FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (M8(IM)),
     XWHERE IS, IF NONE, GO TO 9999
C
      IF (NFAIL(IS)) NE (0), GO TO 30
      CALL CR03(IS)
      LET STATS(IS) = 1
      LET STIME(IS) = ETIME
C
   30 LET NFAIL(IS) = NFAIL(IS) + 1
C
      GO TO 9999
C
    5 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2150),
     XAND (M8(M)) EQ (TRSM), AND (M5(M)) EQ (VB), WHERE IM, IF NONE,
     XGO TO 9999
C
      STORE VC IN M7(IM)
      GO TO 9999
C
 9999 RETURN
      END
*IBFTC R2500
            SUBROUTINE R2500
C
C
C.....PURPOSE - START MAINT. FOR FAILURE FOR UNDETERMINED FAILURE.
C
C
C.....IDSUB = 2500.
C
C
C
      DO TO 1, FOR EACH FLVLS I
      IF (VA) EQ (FLVL(I)), GO TO 2
    1 LOOP
      GO TO 9999
C
    2 FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      CREATE MAINT CALLED M
C
      STORE SERNO(IS) IN M4(M)
      STORE ETIME     IN M1(M)
      STORE VB        IN M5(M)
      STORE TRSM      IN M8(M)
      STORE IDSUB     IN M9(M)
C
```

```
      FILE M IN MQUE
C
 9999 RETURN
      END
*IBFTC R3100
            SUBROUTINE R3100
C
C
C.....PURPOSE - START FLIGHT.
C
C
C.....IDSUB = 3100.
C
C
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      CALL CRD3(IS)
C
      LET STATS(IS) = 0
      LET STIME(IS) = ETIME
C
      CREATE SORTE CALLED S
C
      STORE SERNO(IS) IN S4(S)
      STORE ETIME     IN S1(S)
      STORE BNO(IS)   IN S3(S)
      STORE TRSM      IN S8(S)
C
      FILE S IN SRTQ
C
 9999 RETURN
      END
*IBFTC R3200
            SUBROUTINE R3200
C
C
C.....PURPOSE - END FLIGHT.
C
C
C.....IDSUB = 3200.
C
C
C
      FIND FIRST, FOR EACH M IN SRTQ, WITH (S8(M)) EQ (TRSM),
     XWHERE ISRT, IF NONE, GO TO 9999
C
      FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
     XWHERE IS, IF NONE, GO TO 9999
C
      CALL CRD3(IS)
```

```
C
       LET STATUS(IS) = 2
       LET STIME(IS) = ETIME
C
       STORE ETIME IN S2(ISRT)
C
       CALL CRD2(ISRT)
C
       REMOVE ISRT FROM SRTQ
       DESTROY SORTE CALLED ISRT
C
 9999 RETURN
       END
*IBFTC CRDC
             SUBROUTINE CRDO(IM)
C
C
C.....PURPOSE - OUTPUT MAINTENANCE DATA.
C
C
C.....CALLED BY R200.
C
C
C
       LET DTIME = M1(IM)
       CALL CLK1(DTIME,I1,I2,I3)
       LET IS1 = I1
       LET IS2 = I2
       LET IS3 = I3
C
       LET DTIME = M2(IM)
       CALL CLK1(DTIME,I1,I2,I3)
       LET IE1 = I1
       LET IE2 = I2
       LET IE3 = I3
C
       IF (IS1) EQ (IE1), GO TO 100
C
    1 IF (IS1) EQ (IE1), GO TO 10
       LET IIE1 = IS1
       LET IIE2 = 23
       LET IIE3 = 59
C
       CALL CNVRT(IS1,IS2,IS3,IIE1,IIE2,IIE3,CNVTHR)
C
       LET IHRELP = CNVTHR * 10.0 + .5
       IF (IHRELP) EQ (0), LET IHRELP = 1
       LET IHRMAN = IHRELP * M3(IM)
C
       WRITE ON TAPE 8, IS1,IS2,IS3,IIE1,IIE2,IIE3,
     X                  M3(IM),M4(IM),M5(IM),IHRMAN,M5(IM),IS1,IHRELP,
     X                  M9(IM),0
```

```
      FORMAT (S8,I4,I2,I2,I4,I2,I2,I2,S8,I4,S7,I4,S8,I4,I5,I3,I3,S1,
     X         I5,S1,I1)
C
      LET IS1 = IS1 + 1
      LET IS2 = 0
      LET IS3 = 0
C
      GO TO 1
C
   10 IF (IS2) NE (IE2), GO TO 100
      IF (IS3) NE (IE3), GO TO 100
      GO TO 9999
C
  100 CALL CNVRT(IS1,IS2,IS3,IE1,IE2,IE3,CNVTHR)
C
      LET IHRELP = CNVTHR * 10.0 + .5
      IF (IHRELP) EQ (0), LET IHRELP = 1
      LET IHRMAN = IHRELP * M3(IM)
C
      WRITE ON TAPE 8, IS1,IS2,IS3,IE1,IE2,IE3,
     X                 M3(IM),M4(IM),M5(IM),IHRMAN,M5(IM),IS1,IHRELP,
     X                 M9(IM),0
      FORMAT (S8,I4,I2,I2,I4,I2,I2,I2,S8,I4,S7,I4,S8,I4,I5,I3,I3,S1,
     X         I5,S1,I1)
C
C
 9999 RETURN
      END
*IBFTC CRD2
            SUBROUTINE CRD2(ISRT)
C
C
C.....PURPOSE - OUTPUT SORTIE DATA.
C
C
C.....CALLED BY R3200.
C
C
C
      LET DTIME = S1(ISRT)
      CALL CLK1(DTIME,I1,I2,I3)
      LET IS1 = I1
      LET IS2 = I2
      LET IS3 = I3
C
      LET DTIME = S2(ISRT)
      CALL CLK1(DTIME,I1,I2,I3)
      LET IE1 = I1
      LET IE2 = I2
      LET IE3 = I3
C
      IF (IS1) EQ (IE1), GO TO 100
```

```
C
    1 IF (IS1) EQ (IE1), GO TO 10
      LET IIE1 = IS1
      LET IIE2 = 23
      LET IIE3 = 59
C
      CALL CNVRT(IS1,IS2,IS3,IIE1,IIE2,IIE3,CNVTHR)
C
      LET IHRELP = CNVTHR * 10.0 + .5
      IF (IHRELP) EQ (0), LET IHRELP = 1
C
      WRITE ON TAPE 8, IS1,IS2,IS3,IIE1,IIE2,IIE3,
     X                  S4(ISRT),S3(ISRT),IS1,IHRELP,S8(ISRT),2
      FORMAT (S8,I4,I2,I2,I4,I2,I2,S10,I4,S14,I4,S10,I3,I3,S1,
     X         I5,S1,I1)
C
      LET IS1 = IS1 + 1
      LET IS2 = 0
      LET IS3 = 0
C
      GO TO 1
C
   10 IF (IS2) NE (IE2), GO TO 100
      IF (IS3) NE (IE3), GO TO 100
      GO TO 9999
C
  100 CALL CNVRT(IS1,IS2,IS3,IE1,IE2,IE3,CNVTHR)
C
      LET IHRELP = CNVTHR * 10.0 + .5
      IF (IHRELP) EQ (0), LET IHRELP = 1
C
      WRITE ON TAPE 8, IS1,IS2,IS3,IE1,IE2,IE3,
     X                  S4(ISRT),S3(ISRT),IS1,IHRELP,S8(ISRT),2
      FORMAT (S8,I4,I2,I2,I4,I2,I2,S10,I4,S14,I4,S10,I3,I3,S1,
     X         I5,S1,I1)
C
 9999 RETURN
      END
*IBFTC CR03
              SUBROUTINE CR03(IS)
C
C
C.....PURPOSE - OUTPUT STATUS DATA.
C
C
C.....CALLED BY - R110, R200, R2100, R2150, R2300, R2500, R3100, R3200.
C
C
C
      LET DTIME = STIME(IS)
      CALL CLK1(DTIME,I1,I2,I3)
      LET IS1 = I1
```

```
      LET IS2 = I2
      LET IS3 = I3
C
      LET DTIME = ETIME
      CALL CLK1(DTIME,I1,I2,I3)
      LET IE1 = I1
      LET IE2 = I2
      LET IE3 = I3
C
      IF (IS1) NE (IE1), GO TO 1000
      IF (IS2) NE (IE2), GO TO 1000
      IF (IS3) NE (IE3), GO TO 1000
      GO TO 9999
C
 1000 IF (IS1) EQ (IE1), GO TO 100
C
    1 IF (IS1) EQ (IE1), GO TO 10
      LET IIE1 = IS1
      LET IIE2 = 23
      LET IIE3 = 59
C
      CALL CNVRT(IS1,IS2,IS3,IIE1,IIE2,IIE3,CNVTHR)
C
      LET IHRELP = CNVTHR * 10.0 + .5
      IF (IHRELP) EQ (0), LET IHRELP = 1
C
      WRITE ON TAPE 8, IS1,IS2,IS3,IIE1,IIE2,IIE3,
     X                  STATS(IS),SERNO(IS),IS1,IHRELP,IDSUB,1
      FORMAT (S8,I4,I2,I2,I4,I2,I2,I2,S8,I4,S28,I5,I3,S1,I5,S1,I1)
C
      LET IS1 = IS1 + 1
      LET IS2 = 0
      LET IS3 = 0
C
      GO TO 1
C
   10 IF (IS2) NE (IE2), GO TO 100
      IF (IS3) NE (IE3), GO TO 100
      GO TO 9999
C
  100 CALL CNVRT(IS1,IS2,IS3,IE1,IE2,IE3,CNVTHR)
C
      LET IHRELP = CNVTHR * 10.0 + .5
      IF (IHRELP) EQ (0), LET IHRELP = 1
C
      WRITE ON TAPE 8, IS1,IS2,IS3,IE1,IE2,IE3,
     X                  STATS(IS),SERNO(IS),IS1,IHRELP,IDSUB,1
      FORMAT (S8,I4,I2,I2,I4,I2,I2,I2,S8,I4,S28,I5,I3,S1,I5,S1,I1)
C
 9999 RETURN
      END
*IDEFTC CNVRT
```

```
              SUBROUTINE CNVRT(IS1,IS2,IS3,IE1,IE2,IE3,CNVTHR)
C
C.....PURPOSE - TO CONVERT STOP - START TIME DAY HOUR MINUTES
C                TO DECIMAL HOURS.
C
      LET FS1 = IS1
      LET FS2 = IS2
      LET FS3 = IS3
      LET FE1 = IE1
      LFT FE2 = IE2
      LET FE3 = IE3
C
      LET CNVTHR = (((FE1 * 24.0 * 60.0) + (FE2 * 60.0) + FE3)
     X            -  ((FS1 * 24.0 * 60.0) + (FS2 * 60.0) + FS3))
     X            / 60.0
C
      RETURN
      END
*IBFTC CLK1
              SUBROUTINE CLK1(DTIME,I1,I2,I3)
C
C.....PURPOSE - TO CONVERT DECIMAL DAYS TO DAYS, HOURS, MINUTES.
C
      LET IDY = DPART(DTIME)
      LET IHR = HPART(DTIME)
      LET IMN = MPART(DTIME)
      IF (IMN) NE (60), GO TO 10
      LET IHR = IHR + 1
      LET IMN = 0
   10 IF (IHR) LS (24), GO TO 20
      LET IHR = IHR - 24
      LET IDY = IDY + 1
   20 LET I1 = IDY
      LET I2 = IHR
      LET I3 = IMN
C
      RETURN
      END
*IBFTC MQMO
      REPORT MQMO
*     MAINT START         END  CREW   A/C   SYS    TID   RID   SID IOSUB
      END
                                                                       *
                                                                   1

      END
*IBFTC MQRPG
      REPORT MQRPG(M,IS1,IS2,IS3,IE1,IE2,IE3)
*        * * *   * * *    *    *    *     *     *     *    *
 *    IS1,IS2,IS3,IE1,IE2,IE3,M3(M),M4(M),M5(M),M6(M),M7(M),M8(M),M9(M)
      END
                                                                       *
                                                                     *
```

```
          END
SENTRY              MAIN
1            37
   1   22 0  Z
  23      0 R
  24      1 R   100   23                        100
00                                              (12)
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

```
97
98
99
   25        0   Z
   26        0 R
   27        1 R      10   26                                           10
0                                                                       (I1)
1
2
3
4
5
6
7
8
9
  28        0   Z
  29        0 R
  30    37  0   Z                                                  999.99999
```

Program 4

AIRCRAFT RECOVERY TIME DISTRIBUTION

IV. AIRCRAFT RECOVERY TIME DISTRIBUTION

The information of most general interest is related to the recovery of the entire aircraft (as opposed to system and subsystem recovery). Figure 15 depicts one of the recovery curves that has been developed.

Note that the two halves of the histogram are identical: each cell entry represents a sortie. The left half shows the touchdown time; the corresponding right half shows the type of sortie, but is left blank because PLANET does not distinguish between sortie types. Provisions are available if the user should decide to add this feature later. Below this, the "HOURS" and "P =" lines show the percentage of aircraft recovered at the hour indicated (in this case, 93 percent are recovered 2 hours after touchdown).

The next line contains a number of statistical computations for facilitating analysis. Of these, two are particularly interesting. "AVG = 1.18" shows that average recovery time (for unscheduled maintenance) was just over 1 hour. "TOTAL = 84.72" shows that approximately 85 hours of Operationally Ready (OR) time were lost in recovering aircraft from the effects of the sorties.

A fighter group is somewhat more fortunate than a bomber wing in that many fighter touchdowns require no unscheduled maintenance. This is reflected in the next line, showing that although 227 sorties were flown, 155 required no unscheduled maintenance, for a breakrate of .317. The following line shows the conventional data, except the "SATUR ON INDEX," which is obtained by dividing total man-hours by total elapsed time. This yields the average number of men on the aircraft for unscheduled maintenance during recovery.

The final line is a precaution that all data may not be graphed.

PROGRAM DESCRIPTION

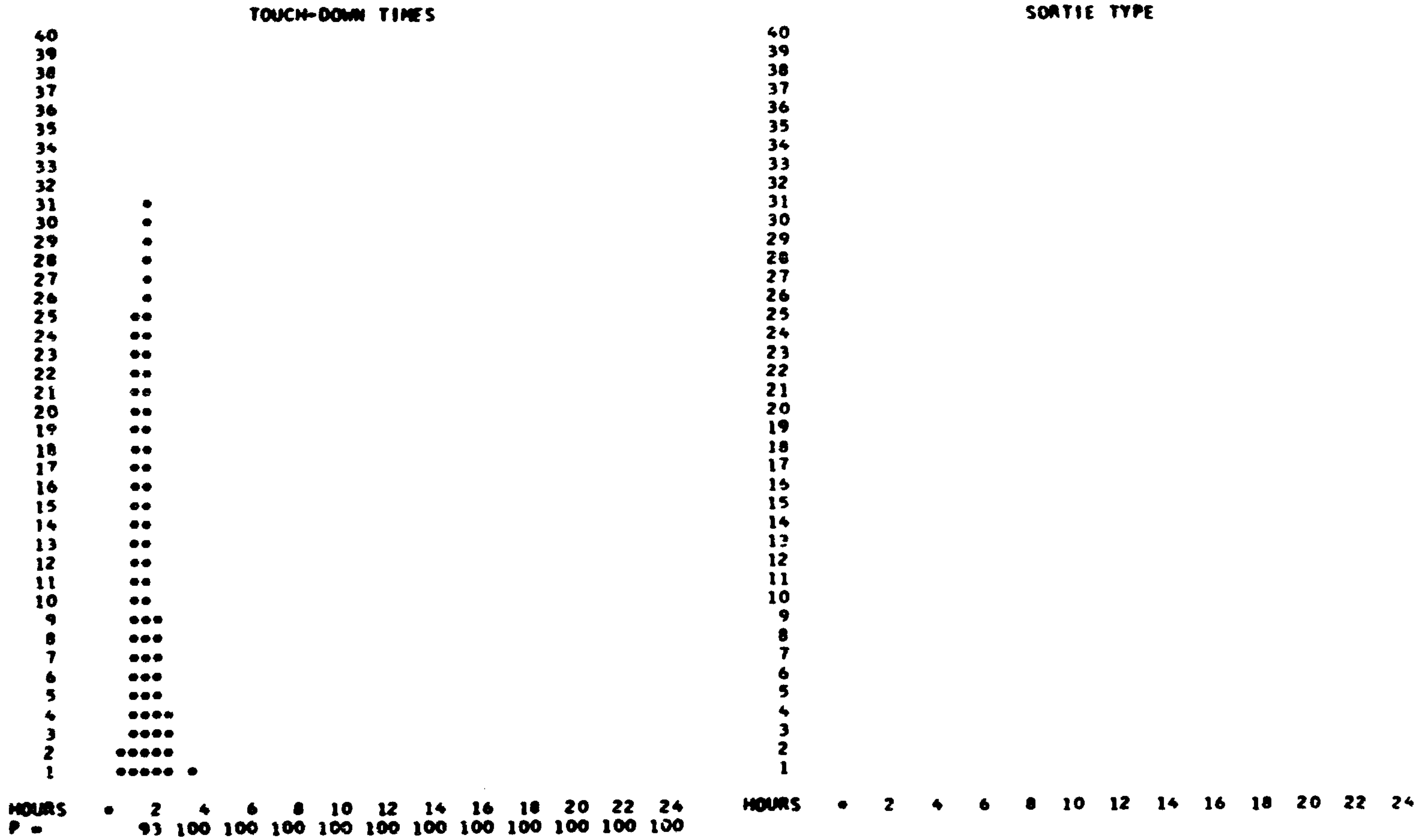
This program is written in standard FØRTRAN IV language.

Input is from card and tape. The highest tail number to be read from tape is read from a card as a five-digit integer. The program

reads the AIRCRAFT RECOVERY TAPE and accumulates the Recovery Time Distributions to be printed.

An Array H contains the histogram of time versus frequency, and an array NO contains a count of frequency at each time. When the last tail number desired has been processed, the histogram is printed.

ENTIRE AIR-CRAFT RECOVERY TIME DISTRIBUTIONS



N = 72 TOTAL TIME = 84.72 AVERAGE = 1.18 MIN.= 0.40 MAX.= 3.27 STD. DEV. = 0.26
TOTAL SORTIES 227 155 SORTIES REQUIRED NO FIX TIME BREAK RATE =0.317
TOTAL MAN HOURS = 250.2 SATURATION INDEX = 3.0 AVE T/A (ALL SORTIES) = 0.4

0 ITEM(S) GREATER THAN 24 HOURS

Fig. 15 -- Aircraft recovery distribution -- an example

```
$IBFTC AR
      DIMENSION H(40,50),NO(50)
      DATA BCAST/1H*/
      DATA BLANK/1H /
      REWIND 8
      READ (5,9000) MAXT
      DO 100 J=1,50
      NO(J)=0
      DO 100 I=1,40
  100 H(I,J)=BLANK
      NSORT=0
      NOFIX=0
      MHRT=0
      NGRT=0
      NSUM=0
      NSD=0
      MIN=999999
      MAX=-99999
  150 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,ITN,MHR,JD,IET,ID,NC
      IF (JD.NE.999) GO TO 250
  200 IF (ITN.GE.MAXT) GO TO 500
      GO TO 150
  250 IF (NC.NE.2) GO TO 150
      NSORT=NSORT+1
      MAINT=0
      MAIN=0
      IST=999999
  300 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,ITN,MHR,JD,IET,ID,NC
      IF (JD.EQ.999) GO TO 375
      IF (NC.NE.0) GO TO 350
      MAIN=1
      IST1=IMN+60*(IHR+24*IDAY)
      JND1=JMN+60*(JHR+24*JDAY)
      IF (ID.EQ.2150) GO TO 325
      IF (ID.EQ.2500) GO TO 325
      IF (ID.EQ.2300) GO TO 325
      IST=JND1
      GO TO 300
  325 IF (MAINT.EQ.0) JND=JND1
      MAINT=1
      IF (IST1.LE.IST) IST=IST1
      IF (JND1.GE.JND) JND=JND1
      MHRT=MHRT+MHR
      GO TO 300
  350 IF (ISC.NE.2) GO TO 300
      IF (MAIN.EQ.0) GO TO 300
  375 IF (MAINT.NE.0) GO TO 400
      NOFIX=NOFIX+1
      GO TO 475
  400 IREC=JND-IST
      IF (MIN.GT.IREC) MIN=IREC
      IF (MAX.LT.IREC) MAX=IREC
```

```
      NSUM=NSUM+IREC
      NSD=NSD+IREC*IREC
      IF (IREC.LE.1440) GO TO 450
      NGRT=NGRT+1
      GO TO 475
  450 J=2+IREC/30
      I=NO(J)+1
      IF (I.GT.40) I=40
      NO(J)=I
      H(I,J)=BCAST
  475 IF (JD.EQ.999) GO TO 200
      GO TO 150
  500 CONTINUE
      N=NSORT-NOFIX
      XN=N
      XSORT=NSORT
      XBR=XN/XSORT
      XSUM=NSUM
      XSUM=XSUM/60.
      XAVG=XSUM/XN
      XSD=NSD
      XSD=XSD/(3600.*XN)-XAVG*XAVG
      XMIN=MIN
      XMAX=MAX
      XMIN=XMIN/60.
      XMAX=XMAX/60.
      XMHR=MHRT
      XMHR=XMHR/10.
      XSI=XMHR/XSUM
      XAT=XSUM/XSORT
      NOS=NO(1)
      DO 550 J=1,49
      NO(J)=100*NOS/N
      NOS=NOS+NO(J+1)
  550 CONTINUE
      WRITE (6,9500)
      WRITE (6,9510)
      DO 600 I=1,40
      I1=41-I
      WRITE (6,9520) I1,(H(I1,J),J=1,50),I1
  600 CONTINUE
      WRITE (6,9530) (I,I=2,24,2),(I,I=2,24,2)
      WRITE (6,9540) (NO(J),J=5,50,4)
      WRITE (6,9550) N,XSUM,XAVG,XMIN,XMAX,XSD
      WRITE (6,9560) NSORT,NOFIX,XBR
      WRITE (6,9570) XMHR,XSI,XAT
      WRITE (6,9580) NGRT
      REWIND 8
      STOP
 9000 FORMAT (I5)
 9100 FORMAT (8X,I4,2I2,I4,3I2,8X,I4,19X,I4,5X,2I3,1X,I5,1X,I1)
 9500 FORMAT (1H1,35X,44HENTIRE AIR-CRAFT RECOVERY TIME DISTRIBUTIONS)
```

```
 9510 FORMAT (1H0,20X,16HTOUCH-DOWN TIMES,50X,11HSORTIE TYPE)
 9520 FORMAT (3X,I2,4X,50A1,6X,I2)
 9530 FORMAT (1H0,5HHOURS,3X,1H*,12I4,5X,5HHOURS,3X,1H*,12I4)
 9540 FORMAT (1X,3HP =,6X,12I4)
 9550 FORMAT (1H0,3HN =,I4,3X,12HTOTAL TIME =,F10.2,3X,9HAVERAGE =,F8.2,
     *         3X,5HMIN.=,F8.2,3X,5HMAX.=,F8.2,3X,11HSTD. DEV. =,F8.2)
 9560 FORMAT (1X,13HTOTAL SORTIES,I5,3X,I5,1X,
     *         28HSORTIES REQUIRED NO FIX TIME,3X,12HBREAK RATE =,F5.3)
 9570 FORMAT (1X,17HTOTAL MAN HOURS =,F8.1,
     *         3X,18HSATURATION INDEX =,F5.1,
     *         3X,23HAVE T/A (ALL SORTIES) =,F5.1)
 9580 FORMAT (1H0,I4,30H ITEM(S) GREATER THAN 24 HOURS)
      END
$ENTRY
   20
$IBSYS          ENDJOB          TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
```

Program 5

SYSTEM RECOVERY

V. SYSTEM RECOVERY

The retrieval of the aircraft system (two-digit) and subsystem (three-digit) data serves several purposes: it enables the monitoring of break and recovery rates; it provides a set of job standards for unscheduled maintenance; and it helps identify aircraft having systems seriously aberrant from the fleet norm.

The two- and three-digit recovery program generally makes a more satisfactory job standard for unscheduled maintenance than does the conventional five-digit method of determining standards. The reason is that to maintenance and control personnel, the "job" consists of the entire action of clearing a complaint. The two- and three-digit recovery program produces the summary of the actions to clear the complaints.

All entries in Fig. 16 show the aircraft serial numbers. The ordinate is a column entry count. The P = line shows the percentage of aircraft recovery by the corresponding time in the hours row. The average recovery follows MEAN = and this is followed by conventional statistical information. Total time shows total OR time lost to this system. SATURATION INDEX is the average number of men working on system recovery.

PROGRAM DESCRIPTION

This program is written in standard FØRTRAN IV language. Note the necessary changes indicated in the program listing to allow the program to run on IBM System 360 computers.

Input is from card and tape. The highest tail number to be read from tape and the highest unit number to be printed are read from a card as five-digit integers. The program reads the AIRCRAFT RECOVERY TAPE as many times as necessary to present a chart for each unit number. The information for as many as twenty units is accumulated each time the tape is read through.

An array ITN contains the list of tail numbers versus time of recovery for as many as twenty units, and an array NO contains a count of number of aircraft at each time for each type of failure. When the last tail number desired has been processed, the charts for each type of failure that occurred in the current set of twenty units are printed and the procedure is resumed for the next set of twenty units.

```
                         SYSTEM RECOVERY

                         UNIT NUMBER     1
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16      10
15      10
14      10
13      10
12      10
11      10
10      10
 9       9
 8       9
 7       9
 6       9
 5       8
 4       8
 3       8
 2       5
 1       1
HOURS LT  3    6    9   12   15   18   21   24   27   30   33   36   39   42   45   48   51   54   57  GT  57
P =     100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100      100

N =  16   TOTAL TIME =      19.30   AVERAGE =     1.21   MIN.=     0.85   MAX.=     1.88   STD. DEV. =     0.09
TOTAL SORTIES  227     135 SORTIES REQUIRED NO FIX TIME   BREAK RATE =0.317
TOTAL MAN HOURS =     57.6   SATURATION INDEX =  3.0   AVE T/A (ALL MAINTENANCE) =  0.1
```

Fig. 16 -- System recovery

```
$IBF+C SR
      DIMENSION ITB(20),NA(20),MHRT(20),MIN(20),MAX(20),NSUM(20),NSD(20)
      DIMENSION ITN(50,20,20),NO(20,20)
      DIMENSION NT1(10),NT2(10),NT3(10),NT4(10)
      DATA NLANK/1H /
      DATA IZ1/4H   0/
      DATA IZ2/4H  00/
      DATA IZ3/4H 000/
      DATA NT1(1)/4H0000/,NT1(2)/4H0001/,NT1(3)/4H0002/,NT1(4)/4H0003/
      DATA NT1(5)/4H0004/,NT1(6)/4H0005/,NT1(7)/4H0006/,NT1(8)/4H0007/
      DATA NT1(9)/4H0008/,NT1(10)/4H0009/
      DATA NT2(1)/4H0000/,NT2(2)/4H0010/,NT2(3)/4H0020/,NT2(4)/4H0030/
      DATA NT2(5)/4H0040/,NT2(6)/4H0050/,NT2(7)/4H0060/,NT2(8)/4H0070/
      DATA NT2(9)/4H0080/,NT2(10)/4H0090/
      DATA NT3(1)/4H0000/,NT3(2)/4H0100/,NT3(3)/4H0200/,NT3(4)/4H0300/
      DATA NT3(5)/4H0400/,NT3(6)/4H0500/,NT3(7)/4H0600/,NT3(8)/4H0700/
      DATA NT3(9)/4H0800/,NT3(10)/4H0900/
      DATA NT4(1)/4H0000/,NT4(2)/4H1000/,NT4(3)/4H2000/,NT4(4)/4H3000/
      DATA NT4(5)/4H4000/,NT4(6)/4H5000/,NT4(7)/4H6000/,NT4(8)/4H7000/
      DATA NT4(9)/4H8000/,NT4(10)/4H9000/
C   DELETE EVERYTHING BETWEEN THE ASTERISKS IF THE PROGRAM IS TO BE RUN
C   ON A MACHINE THAT ALLOWS FEWER THAN FIVE HOLLERITH CHARACTERS/WORD.
C   THESE CARDS SHOULD BE REMOVED TO RUN ON ANY IBM SYSTEM 360 COMPUTER.
C*****************************************************************
      DATA IZ0/5H0000 /
      DO 1 I=1,10
      NT1(I)=NT1(I)-IZ0
      NT2(I)=NT2(I)-IZ0
      NT3(I)=NT3(I)-IZ0
      NT4(I)=NT4(I)-IZ0
    1 CONTINUE
C*****************************************************************
      MIT=20
      READ (5,9000) MAXT,MAXU
      NIT=-MIT
   50 REWIND 8
      NIT=NIT+MIT
      IF (NIT.GT.MAXU) STOP
      DO 100 K=1,MIT
      ITB(K)=0
      MHRT(K)=0
      NSUM(K)=0
      NSD(K)=0
      NA(K)=0
      MIN(K)=999999
      MAX(K)=-99999
      DO 100 J=1,20
      NO(J,K)=0
      DO 100 I=1,50
  100 ITN(I,J,K)=NLANK
      NSORT=0
      NOFIX=0
```

```
      NGRT=0
  150 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,IT4,IT3,IT2,IT1,
     *               MHR,NUF,JD,IET,ID,NC
      ITA=IT1+10*(IT2+10*(IT3+10*IT4))
      MAINT=0
      NSOR=0
      IF (JD.NE.999) GO TO 250
  200 IF (ITA.GE.MAXT) GO TO 500
      GO TO 150
  250 IF (NC.EQ.0) GO TO 310
      IF (NC.NE.2) GO TO 150
      NSORT=NSORT+1
      NSOR=1
      MAIN=0
  300 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,IT4,IT3,IT2,IT1,
     *               MHR,NUF,JD,IET,ID,NC
      IF (JD.EQ.999) GO TO 375
      IF (NC.NE.0) GO TO 350
  310 CONTINUE
      MAIN=1
      IF (ID.EQ.2150) GO TO 325
      IF (ID.EQ.2500) GO TO 325
      IF (ID.EQ.2300) GO TO 325
      IF (ID.EQ.110) GO TO 325
      GO TO 340
  325 MAINT=1
      K=NUF-NIT
      IF (K.LE.0) GO TO 340
      IF (K.GT.MIT) GO TO 340
      ITNO=NT1(IT1+1)+NT2(IT2+1)+NT3(IT3+1)+NT4(IT4+1)
      IF (ITA.GE.10) GO TO 330
      ITNO=IZ1-ITNO
      GO TO 335
  330 IF (ITA.GE.100) GO TO 332
      ITNO=IZ2-ITNO
      GO TO 335
  332 IF (ITA.GE.1000) GO TO 335
      ITNO=IZ3-ITNO
  335 CONTINUE
      NA(K)=NA(K)+1
      MHRT(K)=MHRT(K)+MHR
      IST1=IMN+60*(IHR+24*IDAY)
      JND1=JMN+60*(JHR+24*JDAY)
      IREC=JND1-IST1
      IF (MIN(K).GT.IREC) MIN(K)=IREC
      IF (MAX(K).LT.IREC) MAX(K)=IREC
      NSUM(K)=NSUM(K)+IREC
      NSD(K)=NSD(K)+IREC*IREC
      J=IET/30
  338 J=J+1
      IF (J.GT.20) J=20
      I=NO(J,K)+1
```

```
      IF (I.LE.50) GO TO 339
      IF (J.LT.20) GC TO 338
  339 CONTINUE
      NO(J,K)=I
      ITN(I,J,K)=ITNO
      ITB(K)=ITB(K)+1
  340 IF (NSOR.EQ.0) GO TO 150
      GO TO 300
  350 IF (ISC.NE.2) GO TO 300
      IF (MAIN.EQ.0) GO TO 300
  375 IF (MAINT.EQ.0) NOFIX=NOFIX+1
      IF (JD.EQ.999) GO TO 2C0
      GO TO 150
  500 CONTINUE
      N=NSORT-NOFIX
      XN=N
      XSORT=NSORT
      XBR=XN/XSORT
      DO 700 K=1,MIT
      NUT=NIT+K
      IF (ITB(K).EQ.0) GO TO 700
      N=NA(K)
      XN=N
      XSUM=NSUM(K)
      XSUM=XSUM/60.
      XAVG=XSUM/XN
      XSD=NSD(K)
      XSD=XSD/(3600.*XN)-XAVG*XAVG
      XMIN=MIN(K)
      NMAX=MAX(K)
      XMAX=NMAX
      IF (NMAX.GT.60) NMAX=60
      NMAX1=NMAX/3
      XMIN=XMIN/60.
      XMAX=XMAX/60.
      XMHR=MHRT(K)
      XMHR=XMHR/10.
      XSI=XMHR/XSUM
      XAT=XSUM/XSORT
      NOS=NO(1,K)
      DO 550 J=1,20
      NO(J,K)=100*NOS/N
      NOS=NOS+NO(J+1,K)
  550 CONTINUE
      WRITE (6,9500) NUT
      DO 600 I=1,50
      I1=51-I
      WRITE (6,9520) I1,(ITN(I1,J,K),J=1,20)
  600 CONTINUE
      WRITE (6,9530) (I,I=3,57,3)
      WRITE (6,9540) (NO(J,K),J=1,20)
      WRITE (6,9550) N,XSUM,XAVG,XMIN,XMAX,XSD
```

```
      WRITE (6,9560) NSORT,NOFIX,XBR
      WRITE (6,9570) XMHR,XSI,XAT
  700 CONTINUE
      GO TO 50
 9000 FORMAT (2I5)
 9100 FORMAT (8X,I4,2I2,I4,3I2,8X,4I1,19X,I4,I5,2I3,1X,I5,1X,I1)
 9500 FORMAT (1H1,35X,15HSYSTEM RECOVERY,//,35X,11HUNIT NUMBER,I5)
 9520 FORMAT (3X,I2,4X,20(1X,A4))
 9530 FORMAT (1H0,8HHOURS LT,19(1X,I4),8H  GT  57)
 9540 FORMAT (1X,3HP =,5X,19I5,3X,I5)
 9550 FORMAT (1H0,3HN =,I4,3X,12HTOTAL TIME =,F10.2,3X,9HAVERAGE =,F8.2,
     *        3X,5HMIN.=,F8.2,3X,5HMAX.=,F8.2,3X,11HSTD. DEV. =,F8.2)
 9560 FORMAT (1X,13HTOTAL SORTIES,I5,3X,I5,1X,
     *        28HSORTIES REQUIRED NO FIX TIME,3X,12HBREAK RATE =,F5.3)
 9570 FORMAT (1X,17HTOTAL MAN HOURS =,F8.1,
     *        3X,18HSATURATION INDEX =,F5.1,
     *        3X,27HAVE T/A (ALL MAINTENANCE) =,F5.1)
      END
$ENTRY
   20   15
$IBSYS          ENDJOB          TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
```

Program 6

WORK CENTER RECOVERY

VI. WORK CENTER RECOVERY

THE RECOVERY PROGRAM

The work-center data the recovery program produces are: the touchdown time, the time the work center began the first job and ended the last job, the number of people at work in each ½-hour trial period, as well as the conventional AFM 66-1 data. In plotting the data, touchdown time is set to zero, and the data are plotted from this point in time. The program computes not only the percentage of aircraft still not recovered each ½-hour period, but also the percentage of aircraft being worked on during each period.

Figure 17 is a plot of one month's data. We see that 227 sorties were flown, resulting in 10 requests for Work Center number 1. The matrix shows how these 10 demands were met: one hour after touchdown there were 10 times when a 3-man team was required, etc. By following along the two lines PCB and PBF, we can compare the percentage of aircraft yet to be recovered (PCB) with those actually being worked on (PBF) at each point in time.

Among the facts revealed are that, although 80 percent of the aircraft are recovered within 1 hour after touchdown, by 1½ hours, 30 percent are still being worked on. Note, too, that the work center may not work on 100 percent of the aircraft at all times.

It is worthwhile to point out that a number of analyses become possible with work-center data of the type described. Because touchdown times are on each card, we can compare periods of light and exceedingly heavy loads, thus determining the impact of load on the work center. We can isolate special exercises for similar comparison. We can determine the effects of time of day, or day of week. Further, we can isolate specific sorties in which the aircraft was not touched for, say, more than six hours, and determine why by referring to the Aircraft Recovery time distributions (Program 4) and the System Recovery Program (Program 5). Thus we begin to get at the heart of those factors (manning, scheduling, and performance) that affect the operationally ready rates.

```
      UNIT   1  FLIGHT-LINE DEMANDS FROM TIME OF TOUCH-DOWN (TD=0 HOURS)

      00       01       02       03       04       05       06       07       08       09       10       11       12+
 15
 14
 13
 12
 11
 10
  9
  8
  7
  6
  5
  4
  3   10
  2
  1
MEAN  3.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0

PCB=  100 100  80  30   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
PBF=  100 100  80  30   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

 227 SORTIES FLOWN
  10 REQUESTS ON UNITS
 THE PROBABILITY OF BEING REQUESTED IS 0.04
 TOTAL MAN-HOURS =  37.2
 (PCB=PERCENT OF CRIPPLED BIRDS STILL SICK.  PBF=PERCENT OF BIRDS BEING FIXED.)
```

Fig. 17 -- Flight-line demands from time of touch-down

FLIGHT PROGRAM DESCRIPTION

This program is written in standard FØRTRAN IV language. Note the necessary changes indicated in the program listing to allow the program to run on IBM System 360 computers.

Input is from card and tape. The highest tail number to be read from tape and the highest unit number to be printed are read from a card as five-digit integers. The program reads the AIRCRAFT RECOVERY TAPE as many times as necessary to present a chart for each unit number. The information for as many as twenty units is accumulated each time the tape is read through.

An array ITN contains the counts of crew size versus response time for as many as twenty units, array ZMN contains the mean crew size versus time for each unit, and arrays NCB and NFX contain counts of disabled aircraft and aircraft being fixed at each time for each unit. When the last tail number desired has been processed, the charts for each type of failure that occurred in the current set of twenty units are printed and the procedure is resumed for the next set of twenty units.

```
$IBFTC FD
      DIMENSION ITN(15,25,20),NCB(25,20),NFX(25,20)
      DIMENSION ZNO(25,20),ZMN(25,20),NREQ(20),MHRT(20)
      DIMENSION NT1(10),NT2(10),NT3(10),NT4(10)
      DATA NLANK/1H /
      DATA IZ1/4H   0/
      DATA IZ2/4H  00/
      DATA IZ3/4H 000/
      DATA NT1(1)/4H0000/,NT1(2)/4H0001/,NT1(3)/4H0002/,NT1(4)/4H0003/
      DATA NT1(5)/4H0004/,NT1(6)/4H0005/,NT1(7)/4H0006/,NT1(8)/4H0007/
      DATA NT1(9)/4H0008/,NT1(10)/4H0009/
      DATA NT2(1)/4H0000/,NT2(2)/4H0010/,NT2(3)/4H0020/,NT2(4)/4H0030/
      DATA NT2(5)/4H0040/,NT2(6)/4H0050/,NT2(7)/4H0060/,NT2(8)/4H0070/
      DATA NT2(9)/4H0080/,NT2(10)/4H0090/
      DATA NT3(1)/4H0000/,NT3(2)/4H0100/,NT3(3)/4H0200/,NT3(4)/4H0300/
      DATA NT3(5)/4H0400/,NT3(6)/4H0500/,NT3(7)/4H0600/,NT3(8)/4H0700/
      DATA NT3(9)/4H0800/,NT3(10)/4H0900/
      DATA NT4(1)/4H0000/,NT4(2)/4H1000/,NT4(3)/4H2000/,NT4(4)/4H3000/
      DATA NT4(5)/4H4000/,NT4(6)/4H5000/,NT4(7)/4H6000/,NT4(8)/4H7000/
      DATA NT4(9)/4H8000/,NT4(10)/4H9000/
C   DELETE EVERYTHING BETWEEN THE ASTERISKS IF THE PROGRAM IS TO BE RUN
C   ON A MACHINE THAT ALLOWS FEWER THAN FIVE HOLLERITH CHARACTERS/WORD.
C   THESE CARDS SHOULD BE REMOVED TO RUN ON ANY IBM SYSTEM 360 COMPUTER.
C****************************************************
      DATA IZO/5H0000 /
      DO 1 I=1,10
      NT1(I)=NT1(I)-IZO
      NT2(I)=NT2(I)-IZO
      NT3(I)=NT3(I)-IZO
      NT4(I)=NT4(I)-IZO
    1 CONTINUE
C****************************************************
      MIT=20
      READ (5,9000) MAXT,MAXU
      NIT=-MIT
   50 REWIND 8
      NIT=NIT+MIT
      IF (NIT.GT.MAXU) STOP
      DO 100 K=1,MIT
      MHRT(K)=0
      NREQ(K)=0
      DO 100 J=1,25
      ZNO(J,K)=0.0
      ZMN(J,K)=0.0
      NCB(J,K)=0
      NFX(J,K)=0
      DO 100 I=1,15
  100 ITN(I,J,K)=0
      NSORT=0
  150 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,ITA,
     *                 MHR,MUF,JD,IET,ID,NC
      IF (JD.NE.999) GO TO 250
```

```
200 IF (ITA.GE.MAXT) GO TO 500
    GO TO 150
250 IF (NC.NE.2) GO TO 150
    NSORT=NSORT+1
    MAIN=0
    ITD=JMN+60*(JHR+24*JDAY)
300 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,ITA,
   *                MHR,NUF,JD,IET,ID,NC
    IF (JD.EQ.999) GO TO 200
    IF (NC.NE.0) GO TO 350
310 CONTINUE
    MAIN=1
    IF (ID.EQ.2150) GO TO 325
    IF (ID.EQ.2500) GO TO 325
    IF (ID.EQ.2300) GO TO 325
    IF (ID.EQ.110) GO TO 325
    GO TO 300
325 K=NUF-NIT
    IF (K.LE.0) GO TO 300
    IF (K.GT.MIT) GO TO 300
    NREQ(K)=NREQ(K)+1
    MHRT(K)=MHRT(K)+MHR
    IST1=IMN+60*(IHR+24*IDAY)
    JND1=JMN+60*(JHR+24*JDAY)
    IREC=IST1-ITD
    J=IREC/30+1
    J1=J+IET/5
    IF (J.GT.25) J=25
    IF (J1.GT.25) J1=25
    I=ISC
    IF (I.GT.15) I=15
    ITN(I,J,K)=ITN(I,J,K)+1
    ZNO(J,K)=ZNO(J,K)+1.0
    ZI=I
    ZMN(J,K)=ZMN(J,K)+ZI
    DO 330 J2=1,J1
330 NCB(J2,K)=NCB(J2,K)+1
    DO 332 J2=J,J1
    NFX(J2,K)=NFX(J2,K)+1
332 NOC(J2,K)=NOC(J2,K)-1
    GO TO 300
350 IF (ISC.NE.2) GO TO 300
    IF (MAIN.EQ.0) GO TO 300
375 IF (JD.EQ.999) GO TO 200
    GO TO 150
500 CONTINUE
    DO 700 K=1,MIT
    NUT=NIT+K
    XMHR=MHRT(K)
    XMHR=XMHR/10.
    N=NREQ(K)
    IF (N.EQ.0) GO TO 700
```

```
      ZN=N
      ZS=NSORT
      ZP=ZN/ZS
      DO 510 J=1,25
      ZMN(J,K)=ZMN(J,K)/ZNO(J,K)
      NCB(J,K)=100*NCB(J,K)/N
      NFX(J,K)=100*NFX(J,K)/N
      DO 510 I=1,15
      L=ITN(I,J,K)
      ITN(I,J,K)=NLANK
      IF (L.EQ.0) GO TO 510
      L4=L/10
      L3=L-10*L4
      ITN1=NT3(L3+1)+NT4(L4+1)
      IF (L.LT.10) ITN1=123-ITN1
      ITN(I,J,K)=ITN1
  510 CONTINUE
      WRITE (6,9500) NUT
      WRITE (6,9510) (NT3(I),I=2,10)
      DO 600 I=1,15
      I1=16-I
      WRITE (6,9520) I1,(ITN(I1,J,K),J=1,25)
  600 CONTINUE
      WRITE (6,9535) (ZMN(J,K),J=1,25,2)
      WRITE (6,9550) (NCB(J,K),J=1,25)
      WRITE (6,9560) (NFX(J,K),J=1,25)
      WRITE (6,9570) NSORT
      WRITE (6,9580) N
      WRITE (6,9590) ZP
      WRITE (6,9600) XMHR
      WRITE (6,9610)
  700 CONTINUE
      GO TO 50
 9000 FORMAT (2I5)
 9100 FORMAT (8X,I4,2I2,I4,3I2,8X,I4,19X,I4,I5,2I3,1X,I5,1X,I1)
 9500 FORMAT (1H1,10X,4HUNIT,I4,2X,29HFLIGHT-LINE DEMANDS FROM TIME,
     *            1X,26HOF TOUCH-DOWN (TD=0 HOURS))
 9510 FORMAT (1H0,9X,2H00,9(6X,A2),6X,2H10,6X,2H11,6X,3H12+)
 9520 FORMAT (1H0,2X,I2,3X,25(2X,A2))
 9530 FORMAT (1H0,7X,25I4)
 9535 FORMAT (1H0,4HMEAN,3X,13(F4.1,4X))
 9550 FORMAT (1H0,4HPCB=,3X,25(1X,I3))
 9560 FORMAT (1X,4HPBF=,3X,25(1X,I3))
 9570 FORMAT (1H0,I6,1X,13HSORTIES FLOWN)
 9580 FORMAT (1X,I6,1X,17HREQUESTS ON UNITS)
 9590 FORMAT (4X,30HTHE PROBABILITY OF BEING REQUESTED IS ,F4.2)
 9600 FORMAT (4X,17HTOTAL MAN-HOURS =,F6.1)
 9610 FORMAT (4X,42H(PCB=PERCENT OF CRIPPLED BIRDS STILL SICK.,
     *            2X,34HPBF=PERCENT OF BIRDS BEING FIXED.))
      END
$ENTRY
   20   15   30


$IBSYS          ENDJOB
```

Program 7

<u>FAILURE LIST</u>

VII. FAILURE LIST

The Failure List program provides a visual history of break-rate information. The break-rates are inferred from unscheduled maintenance performed (Fig. 18).

The unscheduled maintenance is produced by having the recovery program search each sortie for unscheduled maintenance (by two- or selected three-digit systems, or both). Each time a fix is encountered, it is recorded under the appropriate system number.

FAILURE LIST

DAY	UNIT	TNO
0.34	3	8
0.34	4	8
0.34	4	1
0.38	3	3
0.42	3	10
0.42	1	10
0.43	5	13
0.55	3	5
0.55	4	3
0.65	2	4
0.65	1	10
0.65	5	14
0.65	1	9
1.34	3	4
1.40	3	8
1.43	2	9
1.57	4	8
1.60	1	10
2.43	2	8
2.65	2	10
3.43	2	9
3.58	3	8
3.65	2	10
3.65	2	8
4.43	2	10
4.55	3	8
6.00	4	3
7.00	2	2
7.43	5	12
7.55	4	8
8.55	3	8
9.35	4	8
9.37	3	8
9.59	1	8
10.34	3	2
10.41	1	8
10.65	2	8
11.34	1	8
11.43	5	12
11.55	3	8
13.00	3	3
14.34	4	2
14.34	3	10
14.38	4	8
14.58	3	10
14.59	3	8
15.43	1	10
15.43	2	9
15.55	1	10
15.65	2	9

Fig. 18

PROGRAM DESCRIPTION

This program is written in standard FØRTRAN IV language.

Input is from card and tape. The times to start and stop printing, the number of bases, and the number of different tail numbers to be listed are read from a card. The times to start and stop contain four characters each, including decimal point; the number of tail numbers (less than 1000) and the number of bases (less than 10) are each read as four-digit integers. If the number of bases is zero or blank, all bases will be listed; otherwise, a card containing the specific base numbers to be listed is read, with each base number entered as a three-digit integer. If the number of specific tail numbers to be listed is zero or blank, all tail numbers will be listed; otherwise, cards each containing 24 three-digit integer numbers identifying the specific tail numbers are read. The program then reads the original PLANET ABC tape and prints the time, unit, and tail number for all failures concerning the specified bases and tail numbers.

```
$IBFTC FL
      DIMENSION IBS(10),ITN(1000),IST(1000),ISTN(1000)
      NT=0
      NPR=0
  100 REWIND 9
      READ (5,9000) TO,TEND,NBASE,NTAIL
      IF (TEND.EQ.0.0) TEND=1.0E10
      IF (NBASE.EQ.0) GO TO 200
      READ (5,9010) (IBS(I),I=1,NBASE)
  200 IF (NTAIL.EQ.0) GO TO 300
      READ (5,9010) (ITN(I),I=1,NTAIL)
  300 READ (9) L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,T
      IF (L2.EQ.3) GO TO 100
      IF (L2.EQ.1900) GO TO 400
      IF (L2.EQ.2150) GO TO 600
      IF (L2.EQ.2300) GO TO 600
      IF (L2.EQ.2500) GO TO 600
  350 IF (L11.LE.0) GO TO 300
      DO 375 I=1,L11
  375 READ (9) I1
      GO TO 300
  400 IF (NBASE.EQ.0) GO TO 450
      DO 425 I=1,NBASE
  425 IF (IBS(I).EQ.L6) GO TO 450
      GO TO 350
  450 IF (L11.LE.0) GO TO 300
      L11=L11-1
      READ (9) I1
      IF (NTAIL.EQ.0) GO TO 500
      DO 475 I=1,NTAIL
  475 IF (ITN(I).EQ.I1) GO TO 500
      GO TO 350
  500 NT=NT+1
      IST(NT)=L10
      ISTN(NT)=I1
      GO TO 350
  600 IF (T.LT.TO) GO TO 350
      IF (T.GT.TEND) GO TO 100
      DO 650 I=1,NT
  650 IF (IST(I).EQ.L10) GO TO 700
      GO TO 350
  700 I1=ISTN(I)
      NPR=NPR-1
      IF (NPR.GT.0) GO TO 800
      NPR=50
      WRITE (6,9500)
  800 WRITE (6,9510) T,L8,I1
      GO TO 350
 9000 FORMAT (2F4.0,2I4)
 9010 FORMAT (24I3/)
 9500 FORMAT (1H1,6X,12HFAILURE LIST,//,4X,15HDAY   UNIT   TNO)
 9510 FORMAT (3X,F6.2,2(2X,I3))
```

```
      END
$ENTRY
 0.0 0.0    0  40
  1  2  3  4  5  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
$IBSYS          ENDJOB          TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
```

Program 8

COST/EFFECTIVENESS

VIII. COST/EFFECTIVENESS

The Cost/Effectiveness program reports the results of dividing the SITE and Logistics costs by the effectiveness. The measure of effectiveness is derived by subtracting the SITE(s) downtime from the total SITE(s) time for the simulated period; site time may be initialized to specify any time period, as well as base, or for the entire fleet of weapons.

INITIALIZATION

Table 5 lists the initialization requirements. Thirty-six arrays are used. Only twelve require values, however.

For initialization formats the user may use the example data (which are strictly hypothetical) contained with the program listing or refer to Part 1, Section II for the initialization instructions.

OUTPUT PROGRAM

The input to the program is the tape generated by the ABC simulator. The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record.

Any change to the personnel, equipment, or spare part quantities, after the initial assignment, causes a message to be printed. The message contains the quantity and type resource, the base, the cost, and the simulated time.

PERMANENT VARIABLES

EBASE - number of bases

SUM - site cost (by base)

PSUM - personnel cost (by base)

ESUM - equipment cost (by base)

SSUM - spare part cost (by base)

SITE - number of site types

COST - cost per site type

QTY - quantity of each site type

PTYPE - number of personnel types

Table 5

VARIABLE DESCRIPTI N AND INITIALIZATION: COST/EFFECTIVENESS

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col. 19-22 (rows)	Array Number of Attribute to Be Entered in Fig. 5 Col. 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1	0	I									Number of bases to be analyzed	EBASE	E	
2-7	1			Z				1						
8	0	I									Number of site types	SITE	E	
9	1		F		V			8			Cost of each site type	COST		A
10	2	I			V			1	8		Quantity of each site type	QTY		A
11	0	I			V						Number of personnel types	PTYPE	E	
12	1		F		V			11			Cost of each personnel type	PCOST		A
13	2	I		Z				1	11			PQTY		A
14	0	I			V						Number of equipment types	ETYPE	E	
15	1		F					14			Cost of each equipment type	ECOST		A
16	2	I		Z				1	14			EQTY		A
17	0	I			V						Number of spare part types	STYPE	E	
18	1		F					17			Cost of each spare part type	SCOST		A
19	2	I		Z				1	17			SQTY		A
20	0		F		V						ENDSIM or end of report	FINSH	E	
21-27	0	I		Z										
28	1		F	Z				1						
29	0		F	Z										
30	1		F	Z				1						
31	0		F	Z										
32	1		F	Z				1						
33	0		F	Z										
34	0		F	Z										
35	0	I			V						Insert code "0" = Base statistics, "1" = Entire fleet	FLEET	E	
36	0	I		Z										

PCOST - cost per personnel type

PQTY - quantity of each personnel type

ETYPE - number of equipment types

ECOST - cost per equipment type

EQTY - quantity of each equipment type

STYPE - number of spare part types

SCOST - cost per spare part type

SQTY - quantity of each spare part type

FINSH - report time

SITID - first variable of detail record

CODE - second variable of label record

BASE - sixth variable of label record

V1 - seventh variable of label record

V2 - eighth variable of label record

V3 - ninth variable of label record

ETIME - twelfth variable of label record

TST - total site time by base

TTST - total site time for fleet

NOR - total site downtime by base

TNOR - total site downtime for fleet

CEFT - cost/effectiveness by base

TCEFT - total cost/effectiveness for fleet

TSUM - total logistics cost for fleet

FLEET - flag to determine if statistics are by base or for fleet

FOLOW - flag to determine if detail record follows label record

SETS

NAME SET USED FOR WORK ENTITIES
    ONE SUBSCRIPT
    RANKED BY SITE ID NUMBER

OWNER SIMSCRIPT SYSTEM

MEMBER WORK: ID = site ID number
    START = downtime of site
    STOP = uptime of site
    CT100 = count of 100 records
    CT200 = count of 200 records

```
C      RECORD DOWN TIME OF SITE
*T WORK 8           T ID     1     I   1EBASE E     I     SET 1  *ID    L
*                   T START  2     F   2FSET  1     I
*                   T STOP   3     F   3LSET  1     I
*                   T CT100  4     I   4SUM   1     F
*                   T CT200  5     I   5PSUM  1     F
*                   T PSET   6     I   6ESUM  1     F
*                   T SSET   7     I   7SSUM  1     F
*                                      8SITE  E     I
*                                      9COST  1     F
*                                     10QTY   2     I
*                                     11PTYPE E     I
*                                     12PCOST 1     F
*                                     13PQTY  2     I
*                                     14ETYPE E     I
*                                     15ECOST 1     F
*                                     16EQTY  2     I
*                                     17STYPE E     I
*                                     18SCOST 1     F
*                                     19SQTY  2     I
*                                     20FINSH 0     F
*                                     21SITID 0     I
*                                     22CODE  0     I
*                                     23BASE  0     I
*                                     24V1    0     I
*                                     25V2    0     I
*                                     26V3    0     I
*                                     27ETIME 0     F
*                                     28TST   1     F
*                                     29TTST  0     F
*                                     30NOR   1     F
*                                     31TNOR  0     F
*                                     32CEFT  1     F
*                                     33TCEFT 0     F
*                                     34TSUM  0     F
*                                     35FLEET 0     I
*                                     36FOLOW 0     I
*IBFTC MAIN
      MAIN ROUTINE
C
C     PLANET - COST/EFFECTIVENESS PROGRAM
C     INPUT - ABC BINARY OUTPUT TAPE
C
      REWIND 9
      LET KFLAG = 0
    1 IF (FOLOW) EQ (1), GO TO 3
C     READ LABEL RECORD
X   2 READ (9) I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
      STORE I2  IN CODE
      STORE I6  IN BASE
      STORE I7  IN V1
      STORE I8  IN V2
      STORE I9  IN V3
      STORE I11 IN FOLOW
      STORE I12 IN ETIME
      GO TO 5
```

```
C
C       READ DETAIL RECORD
X     3 READ (9) I1,I2,I3,I4,I5,I6,I7,I8,F9,F10
        LET FOLOW = 0
        GO TO 2
C
      5 IF (ETIME) GE (FINSH), GO TO 990
        IF (KFLAG) EQ (1), GO TO 8
        IF (ETIME) EQ (0.0), GO TO 8
C       COMPUTE INITIAL LOGISTICS COSTS
        CALL COMPUT
        LET KFLAG = 1
C
C       TEST FOR RECORD TYPE
C
      8 IF (CODE) EQ (    3), GO TO 990
        IF (CODE) EQ (   10), GO TO 10
        IF (CODE) EQ (   20), GO TO 20
        IF (CODE) EQ (   30), GO TO 30
        IF (CODE) EQ (  100), GO TO 100
        IF (CODE) EQ (  200), GO TO 200
        IF (CODE) EQ (1200), GO TO 1200
        IF (CODE) EQ (1210), GO TO 1210
        IF (CODE) EQ (1220), GO TO 1220
        GO TO 1
C
C       INITIAL PERSONNEL QUANTITY
     10 IF (PQTY(BASE,V1)) LS (V3), LET PQTY(BASE,V1) = V3
        GO TO 1
C
C       INITIAL EQUIPMENT QUANTITY
     20 IF (EQTY(BASE,V1)) LS (V3), LET EQTY(BASE,V1) = V3
        GO TO 1
C
C       INITIAL SPARES QUANTITY
     30 IF (SQTY(BASE,V1)) LS (V3), LET SQTY(BASE,V1) = V3
        GO TO 1
C
C       DEMAND ARRIVAL
    100 CALL DOWN
        GO TO 1
C
C       END OF MAINTENANCE
    200 CALL UP
        GO TO 1
C
C       CHANGE IN PERSONNEL
   1200 LET VALUE = FLOAT(V3) * PCOST(V2)
        LET PSUM(BASE) = PSUM(BASE) + VALUE
        CALL PREPT(VALUE)
        GO TO 1
C
C       CHANGE IN SPARE PARTS
   1210 LET VALUE = FLOAT(V3) * SCOST(V2)
        LET SSUM(BASE) = SSUM(BASE) + VALUE
        CALL SREPT(VALUE)
```

```
      GO TO 1
C
C     CHANGE IN EQUIPMENT
 1220 LET VALUE = FLOAT(V3) * ECOST(V2)
      LET ESUM(BASE) = ESUM(BASE) + VALUE
      CALL EREPT(VALUE)
      GO TO 1
C
C     END OF SIMULATION
  990 CALL DONE
      END
*IBFTC COMPUT
      SUBROUTINE COMPUT
C
C     COMPUTE INITIAL PERSONNEL COSTS PER BASE
      DO TO 10, FOR EACH EBASE I
      DO TO 9, FOR EACH PTYPE J
      LET PSUM(I) = PSUM(I) + (FLOAT(PQTY(I,J)) * PCOST(J))
    9 LOOP
   10 LOOP
C
C     COMPUTE INITIAL EQUIPMENT COSTS PER BASE
      DO TO 20, FOR EACH EBASE I
      DO TO 19, FOR EACH ETYPE J
      LET ESUM(I) = ESUM(I) + (FLOAT(EQTY(I,J)) * ECOST(J))
   19 LOOP
   20 LOOP
C
C     COMPUTE INITIAL SPARE PART COSTS PER BASE
      DO TO 30, FOR EACH EBASE I
      DO TO 29, FOR EACH STYPE J
      LET SSUM(I) = SSUM(I) + (FLOAT(SQTY(I,J)) * SCOST(J))
   29 LOOP
   30 LOOP
      RETURN
      END
*IBFTC DOWN
      SUBROUTINE DOWN
C
C     DEMAND ARRIVAL
C     RECORD DOWN TIME OF SITE
C     READ DETAIL RECORD
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,F9,F10
      LET FOLOW = 0
      STORE I1 IN SITID
C
C     TEST TO SEE IF SITE IS ALREADY DOWN
      FIND FIRST, FOR EACH WORK OF SET(BASE), WITH (ID(WORK)) EQ (SITID)
     X, IF NONE, GO TO 10
      LET CT100(WORK) = CT100(WORK) + 1
      RETURN
C
   10 CREATE WORK
      LET ID(WORK) = SITID
      LET START(WORK) = ETIME
```

```
      LET CT100(WORK) = 1
      FILE WORK IN SET(BASE)
      RETURN
      END
*IBFTC UP
      SUBROUTINE UP
C
C     END OF MAINTENANCE
C     RECORD UP TIME OF SITE
C     READ DETAIL RECORD
C
X     READ (9) I1,I2,I3,I4,I5,I6,I7,I8,F9,F10
      LET FOLOW = 0
      STORE I1 IN SITID
C
C     TEST TO SEE IF THIS IS LAST OUTSTANDING DEMAND FOR SITE
      FIND FIRST, FOR EACH WORK OF SET(BASE), WITH (ID(WORK)) EQ (SITID)
      IF (CT100(WORK)) EQ (1), GO TO 10
      LET CT200(WORK) = CT200(WORK) + 1
      IF (CT200(WORK)) LS (CT100(WORK)), RETURN
C
   10 LET STOP(WORK) = ETIME
      LET NOR(BASE) = NOR(BASE) + (STOP(WORK) - START(WORK))
      REMOVE WORK FROM SET(BASE)
      DESTROY WORK
      RETURN
      END
*IBFTC DONE
      SUBROUTINE DONE
C
C     COMPUTE SITE COSTS AND TOTAL SITE TIME PER BASE
C
      DO TO 10, FOR EACH EBASE I
      LET TOTAL = 0.0
      DO TO 9, FOR EACH SITE J
      LET SUM(I) = SUM(I) + (FLOAT(QTY(I,J)) * COST(J))
      LET TOTAL = TOTAL + FLOAT(QTY(I,J))
    9 LOOP
      LET TST(I) = TOTAL * FINSH
   10 LOOP
C
C     COMPUTE NOR TIME FOR OUTSTANDING DEMANDS
C
      DO TO 20, FOR EACH EBASE I
   19 IF SET(I) IS EMPTY, GO TO 20
      REMOVE FIRST WORK FROM SET(I)
      LET NOR(I) = NOR(I) + (FINSH - START(WORK))
      DESTROY WORK
      GO TO 19
   20 LOOP
C
C     TEST TO SEE IF STATISTICS ARE BY BASE OR FOR ENTIRE FLEET
      IF (FLEET) NE (0), CALL SUMRY
C
      DO TO 30, FOR EACH EBASE I
      LET CEFT(I) =(SUM(I)+PSUM(I)+ESUM(I)+SSUM(I))/ (TST(I)-NOR(I))
```

```
   30 LOOP
      CALL OUTPUT
C
      REWIND 9
      CALL EXIT
      END
*IBFTC SUMRY
      SUBROUTINE SUMRY
C
C     COMPUTE STATISTICS FOR ENTIRE FLEET
C
      DO TO 10, FOR EACH EBASE I
      LET TSUM = TSUM + SUM(I)+PSUM(I)+ESUM(I)+SSUM(I)
      LET TNOR = TNOR + NOR(I)
      LET TTST = TTST + TST(I)
   10 LOOP
C
      LET TCEFT = TSUM / (TTST-TNOR)
      CALL TOUT
      REWIND 9
      CALL EXIT
      END
*IBFTC PREPT
      REPORT PREPT(VALUE)
X           ***  TYPE *** PERSONNEL  AT BASE ***       COSTING ******.**
 X          V3        V2                    BASE                   VALUE
            END
                                                                       X
                                   TIME = ***.*****                 1 X
                                           ETIME
      END
*IBFTC EREPT
      REPORT EREPT(VALUE)
X           ***  TYPE *** EQUIPMENT  AT BASE ***       COSTING ******.**
 X          V3        V2                    BASE                   VALUE
            END
                                                                       X
                                   TIME = ***.*****                 1 X
                                           ETIME
      END
*IBFTC SREPT
      REPORT SREPT(VALUE)
X           ***  TYPE *** SPARE PART AT BASE ***       COSTING ******.**
 X          V3        V2                    BASE                   VALUE
            END
                                                                       X
                                   TIME = ***.*****                 1 X
                                           ETIME
      END
*IBFTC OUTPUT
      REPORT OUTPUT
X                               C O S T / E F F E C T I V E N E S S   F O
 X
X                                                          BASE ***   =
 X                                                                 I
  X                             FOR EACH EBASE I
```

```
                                        END
 R    ***.*****    S I M U L A T E D   D A Y S                                 X
        FINSH                                                              5
  ******.**
   CEFT(I)
                                                                           2 X
                           END
 *IBFTC TOUT                                                                   X
       REPORT TOUT
 X
  X                                     C O S T / E F F E C T I V E N E S S   F O
 X
  X
                                                               FLEET  =  **
                                       END
 R    ***.*****    S I M U L A T E D   D A Y S                                 X
        FINSH                                                              5
 ****.**
 TCEFT
                                                                           X
                          END
 $ENTRY                  MAIN
 1               36
    1           0 R
    2      7    1  Z      2     1                             2              BASES
    8           0 R
    9           1 R       2     8                             2              SITES
     50.00                                                    (D6.2)         COST
     25.00
   10           2 R       2     1     2     8      R N        2(16)
      0         6
      4         4
   11           0 R
   12           1 R       6    11                             6              PERSON
     12.00                                                    (D6.2)         PCOST
     12.00
     14.00
     14.00
     16.00
     16.00
  13            2  Z      2     1     6    11
  14            0 R
  15            1 R       4    14                             4              EQUIP
     8.00                                                     (D6.2)         ECOST
     9.00
    10.00
    11.00
 16             2  Z      2     1     4    14
 17             0 R
 18             1 R       5    17                             5              SPARES
     1.00                                                     (D6.2)         SCOST
     2.00
     3.00
     4.00
     5.00
 19             2  Z      2     1     5    17
```

```
20      0 R                                                   25.00000        ENDSIM
21   27 0  Z
28      1  Z     2    1
29      0  Z
30      1  Z     2    1
31      0  Z
32      1  Z     2    1
33      0  Z
34      0  Z
35      0  Z                                                                  FLEET
36      0  Z
              END INITIALIZATION
```

Program 9

BASE SHOPS MAINTENANCE

CAPABILITY

IX. BASE SHOPS MAINTENANCE CAPABILITY

The Base Shops Maintenance Capability program is used to display the outputs from the Bench Repair Simulator.[3] The report consists of five parts: the input to each shop, its output, and he reparable repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of simulation.

An example of the output display is shown in Figs. 19 to 23. Figure 19 is the display of shop statistics, showing the system (or Unit or item) arrivals and departures, reparables in process, and repair cycle times. A separate display is presented for each shop. The example display is for shop No. 3.

The first line of data shows the time at which the statistics were taken. Notice that the report is for day <u>ending</u> 14.000. Since "time" began at time 0.000 in the simulation, and the report is initialized for seven-day periods, the fourteenth day will end at time 14.000 (not 14.999). The next line entry shows the number of reparables that entered the shop (15) for the period and the sum of all reparables entering the shop (30) as of the report period.

The third line entry is the serviceables departing the shop (returned to serviceable stock) (8), during the period, and the sum of all items processed to date (20). The difference between the arrivals and departures is presented as the number in process (10).

The next line displays a distribution of the reparables in process. The average time in process (for the twenty that were processed) is 4.75 days, the maximum time was 10 days, and the minimum 1 day. The standard deviation for the distribution is 2.37.

The repair time distribution is presented both for the period (just 7 days) and accumulated for the fourteen days.

Figure 20 displays the activity Queueing Factors for the period. Column 1 lists the activities in sequence. Columns 2, 3, and 4 list the

distribution of the quantity of reparables processed by each activity for the period. Column 5 lists the average time that the reparables spend in queue behind each activity awaiting some resource. The average queue time is displayed as work time; i.e., off-shift time is not included.

Figure 21 is the Personnel Utilization report. For each personnel type, listed in col. 1., the sum of all personnel on duty for all shifts of the period (of course, the period may be only one shift) is presented in col. 2. The utilization factor, which is the time actually engaged in a process divided by the total duty time available, is presented in col. 3 for each personnel type. The balance of the display is devoted to the man-hours used at each activity for each personnel type. For example, personnel type 2 worked at activities 5, 10, and 11, and a total of 93.25 man-hours were used during the simulation period (seven days).

Figure 22 is the Equipment Utilization report. By equipment type, listed in col. 1, the quantity is listed in col. 2; cols. 3, 4, and 5 list the time the equipment was used, the idle time, and the downtime (all in decimal-days). Note that the summation of these three columns is equal to 14 equipment days for Equipment types 1 and 2, and 21 equipment days for Equipment type 3. This is the total time available for the equipment. Off-shift time is not deducted.

Column 6 is a count of the number of times the shop equipment failed during the period (in this example, 7 days). Column 7 is the utilization factor for the equipment, computed by dividing the total time available (equipment days) into the time in use. Column 8 lists the activities where the equipment was used.

Figure 23 is the display of the Queueing factors for each component spare part. Column 1 lists the spare part ID number. Column 2 lists the quantity or authorized stock level of each spare part. Column 3 lists the number of demands for each spare part during the period.

Columns 4, 5, and 6 list the distribution of the quantity of unfilled demands (average, maximum, and minimum) for each spare part type. Column 7 lists the average queue time--the average time required to fill the demand.

SHOP 3

SYSTEM ARRIVALS, DEPARTURES, IN-PROCESS,
AND REPAIR CYCLE TIMES

DATA FOR PERIOD ENDING DAY 14.000

REPARABLES ENTERING SHOP THIS PERIOD 15, TO DATE 30

SERVICEABLES DEPARTING SHOP THIS PERIOD 8, TO DATE 20

NUMBER IN PROCESS 10

	AVG	MAX	MIN	STD DEV
REPARABLES IN PROCESS	4.75	10	1	2.37
REPAIR TIME				
THIS PERIOD	1.99	3.03	0.96	0.69
TO DATE	1.37	3.03	0.88	0.67

Fig. 19

ACTIVITY QUEUEING FACTORS
FOR PERIOD JUST COMPLETED

ACT. NO.	NO. OF REPS IN QUEUE AVG	MAX	MIN	AVG QUEUE TIME (IN WORK-HOURS)
1	0.33	3.	0.	0.80
2	0.05	3.	0.	0.63
3	0.00	1.	0.	0.00
4	0.05	1.	0.	0.09
5	1.10	5.	0.	0.76
6	0.01	1.	0.	0.02
7	3.18	8.	0.	1.58
8	0.00	1.	0.	0.00
9	4.15	10.	0.	..44
10	0.27	3.	0.	0 26
11	0.00	1.	0.	0.00
12	0.00	2.	0.	0.53
13	0.01	1.	0.	0.01

Fig. 20

PERSONNEL UTILIZATION

PERS TYPE	QTY	UTIL FACT	NO	MAN-HOURS	NO	WORK TIME AT ACTIVITY MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS
1	42	0.09	1	4.41	2	11.82	3	4.83	12	3.01	13	5.79		
2	36	0.32	5	20.40	10	64.10	11	8.75						
3	48	0.27	4	32.03	5	40.79	10	32.05						
4	30	0.40	6	32.22	7	24.18	8	22.85	9	15.63				

Fig. 21

EQUIPMENT UTILIZATION

EQUIPMENT TYPE	QUANTITY	TIME IN USE	IDLE TIME	DOWN TIME	NO. OF FAILURES	UTILIZATION FACTOR	ACTIVITY NOS. WHERE USED
1	2	0.67	13.15	0.18	3	0.048	4
2	2	1.26	12.35	0.39	11	0.090	10
3	3	1.70	19.29	0.01	1	0.081	5

Fig. 22

QUEUEING FACTORS BY COMPONENT SPARES TYPE

REPARABLE QUEUE LENGTHS AND TIMES BY COMPONENTS

COMPONENT TYPE	QUANTITY	DEMANDS THIS PERIOD	NO. OF REPS IN QUEUE AVG	MAX	MIN	AVG QUEUE TIME (IN WORK-HOURS)
1	10	26	0.00	1.	0.	0.00
2	10	20	0.86	3.	0.	0.00
3	8	21	4.30	9.	0.	0.00
4	12	18	0.00	1.	0.	0.00
5	11	27	12.79	21.	6.	0.00

Fig. 23

INITIALIZATION

The Base Shops Maintenance Capability report program requires the initialization of 83 variables. Only 12 require values, however. Table 5, Variable Description and Initialization Table--Base Shop Capability, contains the information required to initialize the report program. An example initialization data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

The input to this program is the binary tape generated by the Bench Repair Simulation Program; this tape is read from logical unit #9.

The input tape consists of 12-word label records with the following format:

Word 1 - irrelevant.

Word 2 - IDD - a four-digit number identifying the "event" or "occurrence" represented by this record.

Words 3, 4, 5 - irrelevant.

Word 6 - INBASE - the base number.

Words 7, 8, 9, 10 - IV1, IV2, IV3, LADDR. These fields are used to store various items of information, depending on the value of IDD.

Word 11 - INDIC - 1 if the next record is a detail record (to be skipped), 0 otherwise.

Word 12 - RTIME - current simulated time.

When a label record is read, the value of INBASE is compared with the constant permanent attribute called BASE; if they are unequal, the record is skipped. (Thus it would require n runs of this analysis program to process all the data from an n-base simulation run, each time changing the value of BASE.)

If the new RTIME is greater than the previous one, subroutine CLOCK is called to check for the end of the operating shift and the end of the report period. If the report period has ended, subroutine ENDPRD is called to generate the reports. Subroutine CLOCK also updates TIME, which is the actual work time elapsed since the beginning of

Table 6

VARIABLE DESCRIPTION AND INITIALIZATION: BASE SHOP CAPABILITY

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col. 19-22 (rows)	Array Number of Attribute to Be Entered in Fig. 5 Col. 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
[illegible]	0			Z										A
[illegible]	0	I			V						Total number of shops	SHOP	E	
[illegible]	1			Z				[illegible]						A
10	0	I			V						Total number of Activities	ACTIV	E	
[illegible]	1			Z				[illegible]						A
[illegible]	[illegible]	I			V						Total number of Personnel Types	[illegible]	E	
[illegible]	[illegible]			Z				[illegible]						A
[illegible]	[illegible]	I			V						Total number of Equipment Types	[illegible]	E	
[illegible]	1			Z				[illegible]						A
[illegible]	[illegible]			Z										A
[illegible]	1			Z				[illegible]						A
[illegible]	[illegible]			Z										A
[illegible]	[illegible]	I			V						[illegible]	[illegible]	E	
[illegible]	[illegible]			Z				[illegible]						A
	[illegible]			Z										A
[illegible]	[illegible]	I			V						[illegible]	[illegible]	[illegible]	
	[illegible]	I			V			[illegible]			[illegible]	[illegible]		A
[illegible]	[illegible]	I			V						[illegible]	[illegible]	E	
[illegible]	1	I			V			[illegible]			[illegible]	[illegible]		A
[illegible]	[illegible]	I		Z				[illegible]	[illegible]		[illegible]	[illegible]		A
[illegible]	[illegible]		F		V						[illegible]	[illegible]		A
[illegible]	[illegible]		F		V						[illegible]	[illegible]		A
[illegible]	[illegible]	I			V						[illegible]	[illegible]		A

simulation. (The automatically defined system variable TIME is used, in order to take advantage of the ACCUMULATE statement.)

Then the appropriate subroutine is called to process the label record. To each significant IDD number, there corresponds a subroutine: e.g., subroutine NEXTAC is called whenever IDD equals 4400. If IDD does not match any of the significant numbers, it is skipped.

If IDD = 3, the end of simulation has been reached; the program terminates after writing the last set of reports.

Error tests intended for the debugging phase have been left in the program, sprinkled throughout. If an error is encountered, this means that something is amiss in this program, in the simulation program, or in the initialization deck. Subroutine ERROR is called, which terminates after outputting the current value of RTIME and a four-letter abbreviation identifying the routine in which the error was detected. For instance, "REA2" refers to the second error condition in subroutine READY.

Subroutine SNAP outputs a "snapshot" of all permanent and temporary variables, as an aid to debugging. The user may insert, at any point, a call to SNAP with an identifier of one to four letters and/or digits; e.g., CALL SNAP (4HNAME). In this example, "NAME" is the identifier. In the current version of the program, ERROR calls SNAP before terminating.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set or last-of-set, and attributes used only to keep track of time in an ACCUMULATE statement such as TQSZA (these always have names beginning with "T").

RTIME - current simulated time; it is obtained from each label record as it is read in.

STIME - the "RTIME" of the previous label record.

TIME - (a variable automatically defined by the system) - number of work days elapsed since the beginning of simulation. For example, suppose there are 40 work-hours in a week. Then if RTIME = 7.0, TIME will be equal to 1.6667 or 1-2/3 (which is 40 divided by 24).

PTIME - the value of "TIME" at the end of the previous report period.

ENDSH - the "RTIME" at which the current shift will end.

ENDPD - the "RTIME" at which the current period will end.

ETIME - the value of "RTIME" at the end of the previous report period.

CURPD - the length in work days (using "TIME") of the period just completed.

CURP - the length in simulated time (using "RTIME") of the period just completed.

CURSH - number of current shift (on a weekly cycle).

CURAC - activity number associated with current label record.

CURSP - spare part number associated with current label record.

CUREP - I.D. number of REP associated with current label record.

SHOP - permanent entity, of which the following are attributes:

RIN - number of reps entering this shop this period.

ROUT - number of reps leaving shop this period.

TRIN - total number of reps in shop (since the beginning of simulation).

TROUT - total number of reps that have left this shop.

MAXR - maximum number of reps in shop this period.

MINR - minimum number of reps this shop this period.

RIP - number of reps currently in process in this shop.

RIPS - a running sum of all the values that RIP has assumed during this period.

RIPSQ - a running sum-square total of all the values that RIP has assumed during this period. E.g., if RIP has had the values 2, 3, 4, 3, 2 in this period, then RIPS is 2 + 3 + 4 + 3 + 2 or 14, and RIPSQ is $2^2 + 3^2 + 4^2 + 3^2 + 2^2$ or 42.

RTS - sum of the repair times of all reps leaving shop this period.

RTSQ - sum of squares of repair times of all reps leaving shop this period.

TRTS - sum of RTS for all periods to date.

TRTSQ - sum of RTSQ for all periods to date.

MXRT - maximum repair time for shop this period.

MNMRT - minimum repair time for shop this period.

TMXRT - maximum repair time for shop, all periods.

TMNRT - minimum repair time for shop, all periods.

ACTIV - activity; a permanent entity, of which the following are attributes:

QSZA - current queue size at this activity.

CQSZA - cumulative total of QSZA, this period.

MXQSA - maximum value of QSZA, this period.

MNQSA - minimum value of QSZA, this period

TIAQS - "time in activity queue, summed;" the total time, in work days, that reps have spent in the queue for this activity.

AVQSA - average queue size at this activity.

AVTAQ - average time in queue for this activity.

AQOUT - number of reps that have left the queue of this activity during this period.

PTYPE - personnel type; a permanent entity, of which the following are attributes:

QTYS - total number of this type of personnel.

CQTY - number of man-days for this personnel type for this period.

ETYPE - equipment type; a permanent entity with the following attributes:

QTYE - total quantity of this equipment type.

NFAIL - number of failures of this type of equipment during this period.

INUSE - quantity of this equipment type currently in use.

CINUS - cumulative total of INUSE, this period.

DOWN - quantity of this type of equipment that is currently down.

CDOWN - cumulative total of DOWN, this period.

SPTYP - spare part type; a permanent entity with the following attributes:

QTYSP - quantity of spares of this type available at beginning of simulation.

DMAND - number of demands for this type of part during this period.

FILL - number of times that such a demand was filled.

QSP - queue size for this type of part.

CQSP - cumulative total of QSP, this period.

MXQSP - maximum value of QSP, this period.

MNQSP - minimum value of QSP, this period.

TISQS - total time, in work days, that reps have spent in the queue for this type of part.

AVQSP - average value of QSP, this period.

AVTSQ - average time in queue for this type of spare part.

RPTYP - rep type; a permanent entity with the following attribute:

SHPNO - number of the shop to which this type of rep belongs.

SHIFT - a permanent entity with the following attribute:

SCHED - 1 if this is a work shift; 0 if this is an off shift.

QTYPR - a permanent attribute with two subscripts:

first subscript: PTYPE

second subscript: SHIFT

meaning: the quantity of personnel of this type, on duty during this shift.

LENSH - the length of a shift.

PEROD - the length of a report period.

BASE - the number of the base for this run; all label records pertaining to any other base will be ignored.

TEMPORARY VARIABLES

REP - a temporary entity with the following attributes:

QTIME - the value of "TIME" when the rep entered the queue for an activity.

BTIME - the value of "RTIME" when the rep entered the system.

IDNO - the I.D. number of the rep: a number obtained from the label record, representing the absolute storage address of the rep in the simulation run.

QFLAG - a number which is equal to zero unless the rep is in the queue for an activity, in which case QFLAG equals the number of that activity.

SFLAG - equal to zero unless rep is in the queue for a spare part, in which case SFLAG equals the number of that type of part.

SPTIM - the value of "TIME" when the rep entered the queue for a spare part.

PLOAD, SLOAD, PACTQ, SACTQ - attributes associated with the sets LOAD and ACTQ.

DUMMY - a temporary entity whose purpose is to save information to be output in Table 4. It has two attributes:

ACNO - the number of an activity at which this type of equipment is to be used.

SSET - successor in the set called "SET."

ENTRY - a temporary entity having to do with the utilization of personnel at different activities. Its attributes are:

ACNO - the number of an activity at which this type of personnel is used.

WKING - number of personnel of this type working at activity whose number equals ACNO.

CWKNG - cumulative total of WKING, this period.

TWKNG - the value of "RTIME" when CWKNG was last updated.

PLIST, SLIST - attributes associated with the set called "LIST."

SETS

LOAD - a set with one subscript, ranked on BTIME.

owner: SHOP

member: REP

The LOAD of each SHOP consists of all the reps that are currently in process in that shop.

ACTQ - a set with one subscript, ranked on BTIME.

owner: ACTIV

member: REP

ACTQ is the queue of all reps currently waiting at an activity.

SET - a FIFO set with one subscript.

owner: ETYPE

member: DUMMY

SET is the set of all activities at which this type of equipment can be used. This information is to be output in Table 4.

LIST - a set with one subscript, ranked on ACNO.

owner: PTYPE

member: ENTRY

LIST has one ENTRY for each activity at which this type of personnel has been used during this report period.

<u>Standard Names for Local Variables</u>

IACNO	always means activity number
IEQNO	always means equipment number
IPERNO	always means personnel number
ISPNO	always means spare part number
IREPNO	always means rep number
ID or IDREP	always means I.D. number of rep
IQTY	always means quantity or number

```
*T REP   8
*                    T CTIME 1      F
*                    T BTIME 2      F
*                    T ICNC  3      I
*                    T CFLAG 41/2   I
*                    T SFLAG 42/2   I
*                    T SPTIM 5      F
*                    T SLCAC 6      I
*                    T SACTC 7      I
*                    T PLCAD 81/2   I
*                    T PACTC 82/2   I
*                                                     LCAC1   *BTIME L
*                                                     ACTQ1   *BTIME L
*T DUMMY2
*                    T SSET  2      I
*                                                     SET 1 *
*T ENTRY8
*                    T ACNC  1      I
*                    T WKING 2      F
*                    T CWKNG 3      F
*                    T TWKNG 4      F
*                    T PLIST 5      I
*                    T SLIST 6      I
*                                                     LIST1   *ACNO  L
*                                   1RTIME          F
*                                   2STIME          F
*                                   3PTIME          F
*                                   4ENCSM          F
*                                   5ENCPC          F
*                                   6CLRPC          F
*                                   7CURSH          I
*                                   8CURAC          I
*                                   9CURSP          I
*                                  10SMCP    E
*                                  11FLCAC   1      I
*                                  12LLCAC   1      I
*                                  13RIN     1      I
*                                  14RCUT    1      I
*                                  15TRIN    1      I
*                                  16TRCUT   1      I
*                                  17MAXR    1      I
*                                  18MINR    1      I
*                                  19RIP     1      I
*                                  20RIFS    1      F
*                                  21RIFSC   1      F
*                                  22RTS     1      F
*                                  23RTSC    1      F
*                                  24TRTS    1      F
*                                  25TRTSC   1      F
*                                  26MXRT    1      F
*                                  27MNRT    1      F
*                                  28TMXRT   1      F
*                                  29TMNRT   1      F
*                                  30ACTIV   E
*                                  31CSZA    1      F
*                                  32CCSZA   1      F
*                                  33TCSZA   1      F
*                                  34MXCSA   1      F
*                                  35MNCSA   1      F
*                                  36TIACS   1      F
```

```
*                                                 32AVCSA 1     F
*                                                 36AVTAC 1     F
*                                                 37FACTC 1     I
*                                                 38LACTC 1     I
*                                                 39ACCUT 1     I
*                                                 40PTYPE E
*                                                 41FLIST 1     I
*                                                 42LLIST 1     I
*                                                 43CTYS  1     I
*                                                 44CCTY  1     F
*                                                 45TCTY  1     F
*                                                 46ETYPE E
*                                                 47CTYE  1     I
*                                                 48NFAIL 1     I
*                                                 50INUSE 1     F
*                                                 51CINUS 1     F
*                                                 52TINUS 1     F
*                                                 53CCWN  1     F
*                                                 54CCCWN 1     F
*                                                 55TCCWN 1     F
*                                                 56FSET  1     I
*                                                 57LSET  1     I
*                                                 60SPTYP E
*                                                 61CTYSP 1     I
*                                                 62DPANC 1     I
*                                                 63FILL  1     I
*                                                 64FSPC  1     I
*                                                 65LSPC  1     I
*                                                 66CSP   1     F
*                                                 67CCSP  1     F
*                                                 68TCSP  1     F
*                                                 69PXCSP 1     F
*                                                 70PNCSP 1     F
*                                                 71TISCS 1     F
*                                                 67AVCSP 1     F
*                                                 71AVTSC 1     F
*                                                 73CUREP       I
*                                                 74CURP        F
*                                                 75ETIME       F
*                                                 76RPTYP E
*                                                 77SHPNC 1     IC
*                                                 78SHIFT E
*                                                 79SCHEC 1     IC
*                                                 80CTYPR 2     F
*                                                 81LENSH       FC
*                                                 82PERCC       FC
*                                                 83BASE        IC
*IBFTC MAIN
            MAIN ROUTINE
            CALL PRELIM
C     ......READ A LABEL RECORD
X  1C       READ (9) K,IDD,K,K,K,INBASE,IV1,IV2,IV3,IADDR,INDIC,T
            LET RTIME = T
C     ......IF THERE IS A DETAIL RECORD, SKIP OVER IT
X           IF (INDIC.EQ.1) READ (9) JUNK
C     ......TERMINATE IF AN ENDSIM RECORD (WITH IDD=3) IS ENCOUNTERED
            IF (IDD) NE (3), GO TO 30
            CALL CLOCK
            LET ENDPD = RTIME
            CALL ENDPRO
```

```
              STOP
C       ......SKIP THIS RECORC IF IT CCES NCT PERTAIN TO THE RIGHT BASE
   3C         IF (INBASE) NE (BASE), GC TO 10
              IF (RTIME) GR (STIME), CALL CLCCK
C       ......CALL THE APPRCPRIATE RCUTINE FCR THIS ICC NUMBER
              IF (IDD) EC (4000), GO TC 50
              IF (IDD) EC (4002), GO TC 52
              IF (ICD) EC (4CC3), GC TC 54
              IF (IDC) EC (4004), GC TC 56
              IF (IDD) EC (4005), GO TC 58
              IF (IDC) EC (4200), GC TC 60
              IF (ICC) EC (1401), GC TC 62
              IF (IDC) EC (4400), GC TC 64
              IF (IDD) EC (4401), GC TC 66
              IF (IDC) EC (4460), GC TC 68
              IF (IDC) EC (4550), GC TC 70
              IF (IDD) EC (4470), GC TC 72
              IF (IDD) EC (4560), GO TC 74
              IF (ICC) EC (4600), GO TC 76
              IF (ICD) EC (470C), GC TO 78
              IF (IDD) EC (4450), GC TC 80
              IF (IDC) EC (4455), GC TC 82
              IF (IDC) EC (4800), GO TC 84
              IF (IDD) EC (48C1), GO TC 86
C       ......FCR ANY CTHER VALUE OF ICC, SKIP THIS RECORC
              GO TC 10
   5C         CALL ACTVTY (IV1)
              GO TC 1C
   52         CALL ECATAC (IV1)
              GC TC 10
   54         CALL PRSNEL (IV1, IV2, IV3)
              GO TC 10
   56         CALL ECUIP (IV1, IV2)
              GC TC 10
   58         CALL SPARES (IV1, IV2)
              GO TC 10
   6C         CALL ARRIV (IV1, IADDR)
              GO TC 10
   62         CALL DEPART (IV1, IADCR)
              GO TO 10
   64         CALL NEXTAC (IV1, IV3, IACCR)
              GC TC 10
   66         CALL READY (IV3)
              GO TO 10
   68         CALL ASINPR (IV1, IV3)
              GC TC 10
   7C         CALL RLESPR (IV1, IV3, IACCR)
              GO TC 1C
   72         CALL ASINEC (IV1, IV3)
              GO TC 10
   74         CALL RLESEC (IV1, IV3)
              GO TO 10
   76         CALL FAIL (IV3)
              GO TO 10
   78         CALL RESTCR (IV3)
              GO TC 10
   8C         CALL SPAVL (IV1)
              GO TO 10
   82         CALL INSPQ (IV1, IV2)
              GC TC 10
```

```
   84       CALL SPRET (IVI)
            GO TC 10
   86       CALL LVSPO (IVI, IADDR)
            GO TC 10
            END
*IBFTC PRELIM
            SUBRCUTINE PRELIM
C     ......INITIALIZE SOME SYSTEM VARIABLES
            LET ENDPD = PERCD
            LET ENDSH = LENSH
            LET CURSH = 1
C     ......INITIALIZE EACH MINIMUM TC A VERY LARGE NUMBER
            DC TC 2C, FOR EACH SHOP I
            LET MNRT(I) = 10C00.
            LET THNRT(I) = 10000.
   2C       LCOP
            RETURN
            END
*IBFTC CLOCK
            SUBROUTINE CLCCK
C     ......THIS ROUTINE KEEPS TRACK CF TIME, ENC-CF-PERIOD, ANC END-OF-
C     ......     SHIFT.  'TIME' IS THE ACTUAL WCRK-TIME ELAPSED SINCE
C     ......     THE BEGINNING CF SIMULATICN, WHEREAS 'RTIME' IS THE
C     ......     CURRENT SIMULATEC TIME.
   2C       LET T = AMIN1 (RTIME, ENCSH, ENCPC)
C     ......UPDATE TIME IF SCME WCRK-TIME HAS ELAPSEC, THAT IS, IF THIS
C     ......     IS A WORKING SHIFT
            IF (SCHED(CURSH)) EC (1), LET TIME = TIME + T - STIME
C     ......UPDATE STIME
            LET STIME = T
            IF (RTIME) EC (T), GC TC 100
C     ......UPDATE THE NO. CF MAN-DAYS (CCTY) FCR EACH PERSONNEL TYPE I
            DO TC 40, FOR EACH PTYPE I
            ACCUMULATE QTYPR(I,CURSH) INTC CCTY(I) SINCE TQTY(I)
   4C       LCOP
C     ......THERE IS AN END-OF-SHIFT ANC/CR AN ENC-OF-PERIOD. DETERMINE
C     ......     WHICH CAME FIRST.
            IF (ENDSH) LE (ENDPD), GC TC 50
C     ......END-OF-PERICD
            CALL ENDPRD
            GO TO 20
C     ......END-OF-SHIFT. UPCATE CURSH ANC ENCSH.
   5C       LET CURSH = MCC (CURSH, NSHIFT) + 1
            LET ENDSH = ENDSH + LENSH
            GO TO 20
  1CC       RETURN
            END
*IBFTC ENDPRD
            SUBROUTINE ENDPRD
C     ......END OF A REPORT PERIOD.
C     ......COMPUTE CUMP AND CUMPD.
            LET S = STIME
            LET CUMP = S - ETIME
            LET ETIME = S
            LET CUMPD = TIME - PTIME
            LET PTIME = TIME
C     ......IF NC WORK-TIME HAS ELAPSEC, CCN'T OUTPUT ANYTHING
            IF (CUMPD) EQ (0.), GC TC 100
C     ......OUTPLT THE REPCRTS FOR THIS PERIOC
            CALL CUT1
```

```
            CALL CLT2
            CALL CLT3
            CALL CLT4
            CALL CLT5
            LET ENDPC = ENCPD + PERCC
  1CC       RETLRN
            END
*IBFTC ACTVTY
            SUBRCUTINE ACTVTY (IACNC)
C     ......THIS RCLTINE IS CALLEC WHEN ICC=4C00. (AT BEGINNING OF RUN)
            IF (IACNC) GR (NACTIV), CALL ERRCR (4HACTV)
            LET CLRAC = IACNC
            RETLRN
            END
*IBFTC ECATAC
            SUBRCLTINE ECATAC (IECNC)
C     ......THIS RCLTINE IS CALLEC WHEN ICC=4C02. (AT BEGINNING OF RLN)
            IF (IECNC) GR (NETYPE), CALL ERRCR (4HECAT)
C     ......SAVE ACTIVITY NUMBERS FCR REPCRT NC. 4
            CREATE DUMMY CALLEC ITEM
            LET ACNC(ITEM) = CURAC
            FILE ITEM IN SET(IECNC)
            RETLRN
            END
*IBFTC PRSNEL
            SLBRCLTINE PRSNEL (IPERNC, ICTY, ISHIFT)
C     ......THIS RCLTINE IS CALLEC WHEN ICC=4003. (AT BEGINNING OF RUN)
            IF (IPERNC) GR (NPTYPE), CALL ERRCR (4HPRSN)
            LET CTYS(IPERNC) = CTYS(IPERNC) + ICTY
            LET QTYPR(IPERNC,ISHIFT) = ICTY
            RETLRN
            END
*IBFTC EQUIP
            SLBRCUTINE ECUIP (IEQNC, ICTY)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4004. (AT BEGINNING OF RUN)
            IF (IECNC) GR (NETYPE), CALL ERRCR (4HEQUI)
            LET QTYE(IEQNC) = ICTY
            RETLRN
            END
*IBFTC SPARES
            SLBRCLTINE SPARES (ISPNC, ICTY)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4005. (AT BEGINNING OF RUN)
            IF (ISPNC) GR (NSPTYP), CALL ERROR (4HSPAR)
            LET QTYSP(ISPNC) = ICTY
            RETLR
            END
*IBFTC ARRIV
            SUBRCLTINE ARRIV (IREPNC, IC)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4200.
C     ......A REP HAS ENTEREC THE SYSTEM. CREATE A TEMPORARY RECORD FOR
C     ......     IT, FILE IT INTO THE APPRCPRIATE SHOP, ANC UPDATE THE
C     ......     STATISTICS FCR THIS SHCP.
            CREATE REP
            LET RTIME(REP) = RTIME
            LET IDNO(REP) = ID
            LET ISHCP = SHPNC(IREPNC)
            LET NEWRIP = RIP(ISHCP) + 1
            LET RIP(ISHCP) = NEWRIP
            LET RIN(ISHCP) = RIN(ISHCP) + 1
            LET MAXR(ISHOP) = MAXO (NEWRIP, MAXR(ISHCP))
```

```
          LET FRIP = NEWRIP
          LET RIPS(ISHOP) = RIPS(ISHOP) + FRIP
          LET RIPSQ(ISHOP) = RIPSQ(ISHOP) + FRIP**2
          FILE REP IN LCAC(ISHOP)
          RETURN
          END
*IBFTC DEPART
          SUBROUTINE DEPART (IREPNO, ID)
C     ......THIS ROUTINE IS CALLED WHEN ICC=1401.
C     ......A REP HAS LEFT THE SYSTEM. REMOVE AND DESTROY IT.
          LET ISHOP = SHPNO(IREPNO)
          FIND FIRST REP, FOR EACH REP IN LCAC(ISHOP), WITH
     *      (IDNO(REP))EQ(ID), WHERE REP, IF NONE, CALL ERROR (4HDEPA)
          REMOVE REP FROM LCAC(ISHOP)
          LET NEWRIP = RIP(ISHOP) - 1
          LET RIP(ISHOP) = NEWRIP
          LET ROUT(ISHOP) = ROUT(ISHOP) + 1
          LET MINR(ISHOP) = MINO (NEWRIP, MINR(ISHOP))
          LET FRIP = NEWRIP
          LET RIPS(ISHOP) = RIPS(ISHOP) + FRIP
          LET RIPSQ(ISHOP) = RIPSQ(ISHOP) + FRIP**2
          LET REPTIM = RTIME - QTIME(REP)
          LET MXRT(ISHOP) = AMAX1 (REPTIM, MXRT(ISHOP))
          LET MNRT(ISHOP) = AMIN1 (REPTIM, MNRT(ISHOP))
          LET RTS(ISHOP) =  RTS(ISHOP) + REPTIM
          LET RTSQ(ISHOP) = RTSQ(ISHOP) + REPTIM**2
          DESTROY REP
          RETURN
          END
*IBFTC NEXTAC
          SUBROUTINE NEXTAC (IREPNO, IACNO, ICREP)
C     ......THIS ROUTINE IS CALLED WHEN ICC=4400.
C     ......A REP IS SUBMITTED (OR RE-SUBMITTED) TO THIS ACTIVITY.
          LET CURAC = IACNO
C     ......DO NOTHING IF ACTIVITY = 0 (RECEIVING)
          IF (IACNO) EQ (0), GO TO 50
          LET CUREP = ICREP
C     ......FIND THE REP BY SEARCHING THE LCAC OF THE APPROPRIATE SHOP.
          FIND FIRST REP, FOR EACH REP IN LCAC(SHPNO(IREPNO)), WITH
     *      (IDNO(REP)) EQ (ICREP), WHERE REP, IF NONE, CALL
     *      ERROR (4HNEXT)
          LET K = QFLAG(REP)
C     ......IF QFLAG EQUALS THE ACT. NO., THIS REP IS ALREADY IN THE
C     ......     QUEUE FOR THIS ACTIVITY, SO DO NOTHING.
          IF (K) EQ (IACNO), GO TO 50
C     ......IF REP IS ALREADY IN SOME OTHER QUEUE, THIS IS AN ERROR.
          IF (K) NE (0), CALL ERROR (4HNEX2)
C     ......IF QFLAG = 0, FILE IT INTO THE QUEUE FOR THIS ACTIVITY.
          LET QFLAG(REP) = IACNO
          ACC QSZA(IACNO) INTO CQSZA(IACNO) SINCE TQSZA(IACNO), ADD 1.
          LET MXQSA(IACNO) = AMAX1 (QSZA(IACNO), MXQSA(IACNO))
          LET QTIME(REP) = TIME
          FILE REP IN ACTQ(IACNO)
   50     RETURN
          END
*IBFTC READY
          SUBROUTINE READY (IACNO)
C     ......THIS ROUTINE IS CALLED WHEN ICC=4401.
C     ......THIS REP IS READY TO BE WORKED ON. REMOVE IT FROM QUEUE FOR
C     ......     THIS ACTIVITY, AND TAKE STATISTICS.
```

```
          ACC QSZA(IACNC) INTC CQSZA(IACNO) SINCE TQSZA(IACNO),ACC -1.
          IF (CSZA(IACNC)) LE (-1.), CALL ERROR (4HREAC)
          FIND FIRST, FOR EACH REP IN ACTQ(IACNO), WITH (ICNO(REP)) EQ
     *      (CUREP), WHERE REP, IF NCNE, CALL ERROR (4HREA2)
          REMOVE REP FRCM ACTQ(IACNC)
          LET MNCSA(IACNC) = AMIN1 (CSZA(IACNC), MNQSA(IACNO))
          LET TIACS(IACNC) = TIACS(IACNC) + TIME - QTIME(REP)
          LET ACCUT(IACNC) = ACCUT(IACNC) + 1
          LET QFLAG(REP) = 0
          RETURN
          END
*IBFTC ASINPR
          SUBRCUTINE ASINPR (IPERNC, ICTY)
C     ......THIS RCUTINE IS CALLED WHEN ICC=4460.
C     ......ASSIGN PERSCNNEL TC AN ACTIVITY.
          LET C = IQTY
          LET R = RTIME
          LET IACNC = CURAC
          IF (IACNC) EC (0), CALL ERRCR (4HASPR)
C     ......IF THIS IS THE FIRST TIME (DURING THIS REPORT PERIOD) THAT
C     ......     PERSONNEL CF THIS TYPE HAVE BEEN ASSIGNEC TO THIS
C     ......     ACTIVITY, CREATE A NEW ENTRY ANC FILE IT INTO LIST FOR
C     ......     THIS PERSCNNEL NC. IN ANY CASE, TAKE STATISTICS.
          FIND FIRST, FCR EACH ENTRY CF LIST(IPERNO), WITH
     *      (ACNC(ENTRY)) EC (IACNC), WHERE ENTRY, IF NONE, GO TO 2C
          LET W = WKING(ENTRY)
          LET CWKNG(ENTRY) = CWKNG(ENTRY) + W * (R-TWKNG(ENTRY))
          LET WKING(ENTRY) = W + C
          GO TC 50
   2C     CREATE ENTRY
          LET ACNC(ENTRY) = IACNC
          LET WKING(ENTRY) = C
          FILE ENTRY IN LIST(IPERNC)
   5C     LET TWKNG(ENTRY) = R
          RETURN
          END
*IBFTC RLESPR
          SUBRCUTINE RLESPR (IPERNC, IGTY, IACNC)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4550.
C     ......PERSCNNEL HAVE BEEN RELEASED FROM THIS ACTIVITY. TAKE
C     ......      STATISTICS.
          FIND FIRST, FOR EACH ENTRY OF LIST(IPERNO), WITH
     *      (ACNC(ENTRY)) EC (IACNC), WHERE ENTRY, IF NONE, CALL
     *      ERROR (4HRLPR)
          LET W = WKING(ENTRY)
          LET R = RTIME
          LET CWKNG(ENTRY) = CWKNG(ENTRY) + W * (R-TWKNG(ENTRY))
          LET TWKNG(ENTRY) = R
          LET WKING(ENTRY) = W - FLCAT(ICTY)
          RETURN
          END
*IBFTC ASINEQ
          SUBRCUTINE ASINEC (IEQNC, ICTY)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4470.
C     ......ASSIGN EQUIPMENT.
          ACCUMULATE INUSE(IEQNO) INTO CINUS(IEQNO) SINCE
     *      TINUS(IEQNC), ACD FLCAT(ICTY)
          RETURN
          END
*IBFTC RLESEQ
```

```
            SLBRCLTIKE RLESEC (IECNC, ICTY)
C     ......THIS RCLTIKE IS CALLEC WHEN ICC=4560.
C     ......RELEASE ECLIPPENT.
            ACCUPULATE INUSE(IECNC) INTO CINUS(IEGNO) SINCE
     *        TINUS(IECNC), ACC -FLCAT(ICTY)
            IF (INLSE(IECNC)) LE (-1.), CALL ERRCR (4HRLEC)
            RETLRN
            END
*IBFTC FAIL
            SLBRCUTIKE FAIL (IECNC)
C     ......THIS RCLTIKE IS CALLEC WHEN ICC=4600.
C     ......ECUIPPENT FAILURE.
            LET NFAIL(IECNC) = NFAIL(IEQNC) + 1
            LET T = RTIPE
            LET CDCWN(IECNC) = CCCWN(IEQNC) + CCWN(IEQNC) *
     *                                             (T - TCOWN(IEQNO))
            LET TDCWN(IECNC) = T
            LET DCWN(IECNC) = DCWN(IECNC) + 1.
            RETLRN
            END
*IBFTC RESTOR
            SLBRCLTIKE RESTCR (IECNC)
C     ......THIS RCLTIKE IS CALLEC WHEN ICC=4700.
C     ......EQUIPPENT RESTORED.
            LET T = RTIPE
            LET CDCWN(IECNC) = CCCWN(IECNC) + CCWN(IEQNO) *
     *                                             (T - TCCWN(IEQNC))
            LET TDCWN(IECNC) = T
            LET DOWN(IECNC) = DOWN(IECNO) - 1.
            IF (DOWN(IEQNC)) LE (-1.), CALL ERRCR (4HREST)
            RETLRN
            END
*IBFTC SPAVL
            SUBRCLTIKE SPAVL (ISPNC)
C     ......THIS RCLTIKE IS CALLEC WHEN ICC=4450.
C     ......THERE IS A DEPAND FCR A SPARE. THE SPARE IS AVAILABLE, SO
C     ......     THE DEPANC IS IPMECIATELY FILLEC.
            LET DMAND(ISPNC) = CMANC(ISPNC) + 1
            LET FILL(ISPNC) = FILL(ISPNC) + 1
            LET MXCSP(ISPNC) = AMAX1 (CSP(ISPNO)+1., MXCSP(ISPNO))
            RETURN
            END
*IBFTC INSPQ
            SLBRCUTIKE INSPC (ISPNC, IREPNC)
C     ......THIS RCUTIKE IS CALLEC WHEN ICC=4455.
C     ......THERE IS A DEPANC FCR AN UNAVAILABLE SPARE PART.
            FIND FIRST, FCR EACH REP IN LCAC(SPPNC(IREPNC)), WITH
     *        (IDNC(REP)) EC (CUREP), WHERE REP, IF NONE, CALL
     *        ERROR (4HINSP)
            LET S = SFLAG(REP)
C     ......IF SFLAG = SPARE PART NC., THIS REP IS ALREACY IN QUEUE FOR
C     ......     THIS PART, SC CC NCTHING.
            IF (S) EC (ISPNC), GC TC 50
C     ......IF REP IS ALREACY IN CUEUE FCR A CIFFERENT PART, CALL ERROR.
            IF (S) NE (0), CALL ERRCR (4HINS2)
C     ......IF SFLAG = 0, PUT IT IN CUEUE FCR THIS SPARE PART NO.
            LET SFLAG(REP) = ISPNC
            LET DMAND(ISPNC) = CMANC(ISPNC) + 1
            ACC CSP(ISPNC) INTC CCSP(ISPNC) SINCE TCSP(ISPNC), ACC 1.
            LET MXCSP(ISPNC) = AMAX1 (CSP(ISPNC), MXCSP(ISPNC))
```

```
          LET SPTIM(REP) = TIME
   5C     RETURN
          END
*IBFTC SPRET
          SUBROUTINE SPRET (ISPNC)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4800.
C     ......(ALL WE NEED FRCM THIS LABEL RECCRC IS THE SPARE PART NO.)
          LET CURSP = ISPNC
          RETURN
          END
*IBFTC LVSPC
          SUBRCUTINE LVSPC (IREPNC, ICREP)
C     ......THIS RCUTINE IS CALLEC WHEN ICC=4801.
C     ......A SPARE PART IS AVAILABLE.
C     ......IF NC REP WAS WAITING FCR THIS PART, CC NOTHING.
          IF (ICREP) EC (0), GC TC 50
C     ......TAKE THIS REP OUT OF THE CUEUE FOR THIS SPARE PART.
          LET ISPNC = CURSP
          LET FILL(ISPNC) = FILL(ISPNO) + 1
          ACC CSP(ISPNC) INTC CQSP(ISPNC) SINCE TQSP(ISPNO), ADC -1.
          IF (QSP(ISPNO)) LE (-1.), CALL ERROR (4HLVSP)
          LET MNQSP(ISPNC) = AMIN1 (CSP(ISPNC), MNQSP(ISPNO))
          FIND FIRST, FOR EACH REP IN LCAC(SHPNC(IREPNC)), WITH
    *       (IDNC(REP)) EC (ICREP), WHERE REP, IF NONE, CALL
    *       ERROR (4HLVS2)
          LET TISCS(ISPNO) = TISCS(ISPNC) + TIME - SPTIM(REP)
          LET SFLAG(REP) = 0
   5C     RETURN
          END
*IBFTC OUT1
          SUBROUTINE CUT1
C     ......GENERATE A 'TABLE 1' REPORT FCR EACH SHOP.
          DO TO 100, FOR EACH SHCP I
          LET IRIN = RIN(I)
          LET IRCUT = ROUT(I)
          LET FRCUT = IRCUT
          LET FRTS = RTS(I)
          LET FRTSC = RTSC(I)
          LET IRIP = RIP(I)
          LET FRIP = IRIP
          LET ITRIN = TRIN(I) + IRIN
          LET TRIN(I) = ITRIN
          LET ITRCUT = TRCUT(I) + IRCUT
          LET TRCUT(I) = ITRCUT
          LET FTRCUT = ITRCUT
          LET TOTAL = IRIN + IRCUT + 1
          CALL STDDEV (TCTAL, RIPS(I), RIPSQ(I), *AVN, *STCVN)
          IF (IRCUT) EQ (0), LET PMRT(I) = 0.
          CALL STDDEV (FRCUT, FRTS, FRTSC, *AVT, *STCVT)
          LET FTRTS = TRTS(I) + FRTS
          LET TRTS(I) = FTRTS
          LET FTRTSQ = TRTSQ(I) + FRTSC
          LET TRTSQ(I) = FTRTSC
          CALL STDDEV (FTRCUT, FTRTS, FTRTSC, *TAVT, *TSTDVT)
          IF (ITRCUT) EQ (0), GC TC 60
          LET TMXRT(I) = AMAX1 (MXRT(I), TMXRT(I))
          LET FTMNRT = AMIN1 (MNRT(I), TMNRT(I))
          LET TMNRT(I) = FTMNRT
          GO TC 80
   6C     LET FTMNRT = 0.
```

```
   8C       CALL TAB1 (I, AVN, STDVN, AVT, STDVT, TAVT, FTMNRT, TSTDVT)
C     ......RESET VARIABLES FOR NEXT REPORT PERIOD.
            LET RIN(I) = 0
            LET ROUT(I) = 0
            LET MAXR(I) = IRIP
            LET MINR(I) = IRIP
            LET RIPS(I) = FRIP
            LET RIPSQ(I) = FRIP * FRIP
            LET RTS(I) = 0.
            LET RTSC(I) = 0.
            LET MXRT(I) = 0.
            LET MNRT(I) = 10000.
  1CC       LOOP
            RETURN
            END
*IBFTC STDDEV
            SUBROUTINE STDDEV (TOTAL, SUM, SUMSC, AVG, STDV)
C     ......ROUTINE TO COMPUTE A MEAN AND STANDARD DEVIATION.
            IF (TOTAL) LE (0.), GO TO 50
            LET AVG = SUM / TOTAL
            LET STDV = SQRT(AMAX1(SUMSC/TOTAL - AVG*AVG, 0.))
            GO TO 100
   5C       LET AVG = 0.
            LET STDV = 0.
  1CC       RETURN
            END
*IBFTC TAB1
     REPORT TAB1 (I, AVN, STDVN, AVT, STDVT, TAVT, FTMNRT, TSTDVT)
X                                                                SHO
 X
X                                                    SYSTEM ARRIVALS, DEP
X                                                         AND REPAIR
X
 X
X                                        REPARABLES ENTERING SHOP THIS PE
 X
X                                        SERVICEABLES DEPARTING SHOP THIS
 X
X
 X
X                                                                AVG
X                                        REPARABLES IN PROCESS   **.**
 X                                                               AVN
X                                        REPAIR TIME
X                                          THIS PERIOD            *.**
 X                                                               AVT
X                                          TO DATE                *.**
 X                                                               TAVT
                            END
                                                                    X
P  *                                                             1
  I
ARTURES, IN-PROCESS,
CYCLE TIMES                                                      1
DATA FOR PERIOD ENDING DAY  **.***                               1
                              ENDPD
RIOD        **, TO DATE  ***                                     1
        RIN(I)        TRIN(I)
 PERIOD     **, TO DATE  ***                                     1
        ROUT(I)       TROUT(I)
```

```
 NUMBER IN PROCESS        **                                             1
                        RIP(I)
   MAX        MIN     STD DEV                                            1
   **          *       *.**                                              1
  MAXR(I)    MINR(I)   STDVN
                                                                         1
  **.**       *.**      *.**                                             1
  MXRT(I)    MNRT(I)   STDVT
  **,**       *.**      *.**
 TMXRT(I)    FTMNRT   TSTDVT
                                    END
*IBFTC OUT2
                 SUBROUTINE OUT2
C       ......GENERATE TABLE 2.
                 LET C = CURPD
                 DO TO 50, FOR EACH ACTIV I
                 ACCUMULATE CSZA(I) INTO CCSZA(I) SINCE TCSZA(I)
                 LET AVGSA(I) = CCSZA(I) / C
                 IF (AQCLT(I)) EQ (0), GO TO 50
                 LET AVTAC(I) = DECHR(TIACS(I)) / FLOAT(AQOUT(I))
   50            LOOP
                 CALL TAB2
                 DO TO 100, FOR EACH ACTIV I
                 LET CCSZA(I) = 0.
                 LET TIACS(I) = 0.
                 LET FQSZA = QSZA(I)
                 LET MXCSA(I) = FCSZA
                 LET MNCSA(I) = FCSZA
                 LET AQCLT(I) = 0
  100            LOOP
                 RETURN
                 END
*IBFTC TAB2
        REPORT TAB2
X                                                              ACTIVITY QUE
X                                                              FOR PERIOD J
X                                                              NO. OF REPS
X                                                     ACT.
X                                                      NO.        AVG     MAX
X                                                      **        *.**     **.
 X                                                      I     AVQSA(I) MXQSA(
  X    FOR EACH ACTIV I, WITH (MXCSA(I)) GR (0.)
                                    END
                                                                          X
LEING FACTORS                                                         1
LST COMPLETED                                                         12
IN QUEUE
                AVG QUEUE TIME
     MIN     (IN WORK-HOURS)                                           1
      *.          **.**
I) MNQSA(I)    AVTAC(I)
                                                                          X
                                    END
*IBFTC OUT3
                 SUBROUTINE OUT3
C       ......GENERATE TABLE 3.
C       ......BEGIN BY WRITING THE HEADING.
                 CALL TB3HED
                 LET S = STIME
                 DO TO 100, FOR EACH PTYPE I, WITH (QSVS(I)) GR (C)
```

```
          LET SUM = 0.
   10     DO TO 20, FOR EACH ENTRY OF LIST(I)
C      ......BRING 'CWKNG(ENTRY)' UP TO DATE BEFORE ADDING IT INTO SUM.
          LET C = CWKNG(ENTRY) + WKING(ENTRY) * (S-TWKNG(ENTRY))
          LET CWKNG(ENTRY) = C
          LET SUM = SUM + C
   20     REPEAT 10
          LET UTIL = SUM / CQTY(I)
          CALL TB3LIN (I, UTIL)
          LET CQTY(I) = 0.
C      ......EMPTY OUT EACH 'LIST' SET.
   30     DO TO 50, FOR EACH ENTRY OF LIST(I)
          REMOVE ENTRY FROM LIST(I)
          DESTROY ENTRY
   50     REPEAT 30
  100     LOOP
          RETURN
          END
*IBFTC TB3HED
       REPORT TB3HED
X                                                                      PERSONNEL
X                                        WORK TIME
X                                       AT ACTIVITY
X      PERS      UTIL      MAN-          MAN-          MAN-          MAN-
X      TYPE  QTY FACT  NO  HOURS     NO  HOURS     NO  HOURS     NO  HOURS
                                   END
                                                                           X
UTILIZATION                                                              2
                                                                           X
                                                                           X
       MAN-          MAN-          MAN-          MAN-          MAN-
  NO  HOURS     NO  HOURS     NO  HOURS     NO  HOURS     NO  HOURS     1
                                   END
*IBFTC TB3LIN
       REPORT TB3LIN (I, UTIL)
       ***  **   *.**
 X     I QTYS(I) UTIL
    9     FOR EACH ENTRY OF LIST(I), WITH (FLIST(I)) NE (0)
X                        **   ***.**  **   ***.**  **   ***.**  **   ***.**
 X                        9(ACNO(ENTRY),DECHR(CWKNG(ENTRY)))
                                   END
                                                                           X
                                                                         X
                                                                           X
                                                                         XX
  **   ***.**  **   ***.**  **   ***.**  **   ***.**  **   ***.**
                                                                           X
                                   END
*IBFTC OUT4
          SUBROUTINE OUT4
C      ......GENERATE TABLE 4.
          CALL TB4HED
          LET CUR = CURP
          DO TO 50, FOR EACH ETYPE I, WITH (CTYE(I)) GR (0)
          LET TOTAL = CUR * FLOAT(CTYE(I))
          LET C = CINUS(I)
          LET UTIL = C / TOTAL
          LET FIDLE = TOTAL - C - CDOWN(I)
          CALL TB4LIN (I, FIDLE, UTIL)
          LET NFAIL(I) = 0
```

```
            LET COCWN(I) = 0.
            LET CINUS(I) = 0.
   5C       LCOP
            RETURN
            END
*IBFTC TB4HED
        REPORT TB4HED
.                                                               EQUIPMENT U
X                  EQUIPMENT                   TIME      ICLE       COWN    N
X                    TYPE      QUANTITY    IN USE      TIME       TIME   FA
                                END
                                                                            X
TILIZATION                                                              1
O. OF  LTILIZATICN  ACTIVITY NCS.
ILLRES    FACTOR     WHERE USEC                                         1
                                ENC
*IBFTC TB4LIN
        REPORT TB4LIN (I, FIDLE, UTIL)
X                    **         **        **.**      **.**      **.**
 X                    I       CTYE(I)  CINUS(I)     FIDLE     COOWN(I)
  12        FOR EACH ITEM CF SET(I)
X
 X
                                END
                                                                            X
  *       *.***                                                          X
NFAIL(I)  LTIL
                                                                           XX
                **  **  **  **  **  **  **  **  **  **  **  **
                12(ACNC(ITEM))
                                END
*IBFTC CUTS
            SLBROUTINE CUTS
C      ......GENERATE TABLE 5.
            LET C = CURPD
            DO TC 50, FCR EACH SPTYP I
            ACCUMULATE CSP(I) INTC CCSP(I) SINCE TQSP(I)
            LET AVCSP(I) = CCSP(I) / C
            IF (FILL(I)) EC (O), GC TC 50
            LET AVTSQ(I) = CECHR(TISCS(I)) / FLCAT(FILL(I))
   5C       LCOP
            CALL TAB5
            DO TC 10C, FOR EACH SPTYP I
            LET CQSP(I) = 0.
            LET TISCS(I) = 0.
            LET FQSP = QSP(I)
            LET MXCSP(I) = FCSP
            LET MNQSP(I) = FCSP
            LET DMAND(I) = 0
            LET FILL(I) = 0
  10C       LCOP
            RETURN
            END
*IBFTC TAB5
        REPORT TAB5
X                                                   QUEUEING FACTORS BY
X                                                 REPARABLE QUEUE LENGTHS
*
X                                  CCMPCNENT                     CEMANCS
X                                    TYPE       CUANTITY    THIS PERICC
```

```
X                                                  *          **           **
 X                                                 I       QTYSP(I)     CHANC(I) AVQSP
  X    FOR EACH SPTYP I, WITH (QTYSP(I)) GR (0)
                                            END
                                                                                    X
COMPONENT SPARES TYPE                                                          12
 AND TIMES BY COMPONENTS                                                       11
 NO. OF REPS IN QUEUE
                                       AVG QUEUE TIME
AVG       MAX       MIN       (IN WORK-HOURS)                                   1
*.**      **.       *.            **.**
(I)    MXQSP(I)  MNQSP(I)       AVTSQ(I)
                                                                                    X
                                            END
*IBFTC ERROR
            SUBROUTINE ERROR (NAME)
            CALL ERREPT (NAME)
            CALL SNAP (4HERROR)
            STOP
            END
*IBFTC ERREPT
      REPORT ERREPT (NAME)
X     JOB TERMINATED AT TIME    **.*** BECAUSE OF ERROR IN SUBROUTINE ***
 X                               RTIME                                           NA
X                                                                             (ABBRE
                                            END
                                                                                    X
A**
ME
VIATION)
                                            END
*IBFTC SNAP
            SUBROUTINE SNAP (LABEL)
            CALL SNP1 (LABEL)
            CALL SNP2 (I), FOR EACH SHOP I
            RETURN
            END
*IBFTC SNP1
      REPORT SNP1 (LABEL)
X                                                                     SNAPSHOT REQUE
 X
X        RTIME       STIME       PTIME       ENCSP       ENCPO       CL
X        *.*****     *.*****     *.*****     *.*****     *.*****     *
 X       RTIME       STIME       PTIME       ENCSP       ENCPO       CL
X    SHOPS - RIN ROUT TRIN TROUT MAXR MINR RIP   RIPS      RIPSQ     RT
X     *     *     *     *     *     *     *     *     *.*      **.*     **
 X     I RIN(I) ROUT(I) TRIN(I) TROUT(I) MAXR(I) MINR(I) RIP(I) RIPS(I
 X
  X    FOR EACH SHOP I
X     ACTIVITIES ---        CSZA              CCSZA              TQSZA
X          *                *.*               *.*                *.*****
 X         I                CSZA(I)           CCSZA(I)           TQSZA(I)
  X    FOR EACH ACTIV I, WITH (MXQSA(I)) NE (0.)
X     EQUIPMENT TYPES ---       NFAIL        INUSE        CINUS        TINUS
X                *              *            *.*          *.*          *.*****
 X               I              NFAIL(I)     INUSE(I)     CINUS(I)     TINUS(
  X    FOR EACH ETYPE I, WITH (QTYE(I)) NE (0)
X     SPARE PART TYPES ---      DHAND        FILL         QSP          CQSP
X                *              *            *            *.*          *.*
 X               I              DHAND(I)     FILL(I)      QSP(I)       CQSP
```

```
   X   FCR EACH SPTYP I, WITH (GTYSP(I)) NE (0)
X
                                    ENC
                                                                              X
STED AT **A*                                                              2
       LABEL
RPD       CURP        ETIME       CURSH   CURAC   CURSP
.*****    *.*****     *.*****       *       *       *                     2
RPC       CLRP        ETIME       CURSH   CURAC   CURSP
S        RTSC       TRTS        TRTSC      PXRT    PNRT   TPXRT   TPNRT
.**      ***.**     **.**       ****.**    *.**    *.**    *.**    *.**
) RIPSQ(I) RTS(I) RTSQ(I) TRTS(I) TRTSC(I) PXRT(I) MNRT(I)
                                                    TMXRT(I) TMNRT(I)
                                                                              X
 MXQSA       MNCSA              TIACS       ACOUT                         X
   *.*         *.*               *.**          *
 MXQSA(I)    MNCSA(I)           TIACS(I)    ACCUT(I)
                                                                              X
       DOWN       CDCWN      TCCWN                                        X
        *.*         *.*     *.*****
I)     DOWN(I)    CDCWN(I)   TDCWN(I)
                                                                              X
      TCSP         MXQSP       MNCSP            TISCS                     X
    *.*****          *.*         *.*             *.**
(I)   TQSP(I)      MXQSP(I)    MNCSP(I)         TISCS(I)
                                                                              X
                                                                          X
                                    ENC
*IBFTC SNP2
     REPORT SNP2 (ISHCP)
  14 FCR EACH REP IN LCAC(ISHCP)
X                     REPS IN SHCP  *
 X                                 ISHCP
X     IDNC     *****    *****    *****    *****    *****    *****    ****
 X             14(IDNC(REP))
X     BTIME    *.***    *.***    *.***    *.***    *.***    *.***    *.**
 X             14(BTIME(REP))
X     QTIME    *.***    *.***    *.***    *.***    *.***    *.***    *.**
 X             14(QTIME(REP))
X     QFLAG      *        *        *        *        *        *
 X             14(QFLAG(REP))
                                    ENC
                                                                              X
                                                                          X
                                                                      1
                                                                              X
*   *****    *****    *****    *****    *****    *****    *****
                                                                              X
*   *.***    *.***    *.***    *.***    *.***    *.***    *.***
                                                                              X
*   *.***    *.***    *.***    *.***    *.***    *.***    *.***
                                                                              X
*     *        *        *        *        *        *        *             2
                                                                              X
                                    ENC
```

```
1            83
   1    9     Z
  10          R
  11   29 1   Z    3   10                                   3              SHOP
  30          R
  31   39 1   Z   13   30                                 13               ACTIV
  40          R
  41   45 1   Z    4   40                                 4                PTYPE
  46          R
  47   48 1   Z    3   46                                 3                ETYPE
  49          Z
  50   57 1   Z    3   46
  58   59     Z
  60          R
  61   71 1   Z    5   60                                 5                SPTYP
  72   75     Z
  76          R
  77      1   R    4   76                                 4                RPTYP
 1 2  2 3                                                    4(12)
  78          R
  79      1   R   21   78                                   21             SHIFT
C10C10C10C10C100C10CCC                                      21(11)
  80      2   Z    4   40   21   78
  81          R
8.0                                                         (F1.1)         LENSH
  82          R
  83          R                                          7.0               PEROC
                                                             1             BASE
```

Program 10

NRTS PROGRAM

X. NRTS PROGRAM

The NRTS (not reparable this station) data display shows the reparables shipped off base for repair. It displays the pipeline time distribution for the reparables. This program is used primarily as an input to the Depot Transportation Simulator when the simulators are operated separately.

An example of the output display is shown in Fig. 24. The reparable ID number is listed in column 1, followed by the quantity that was shipped to the depot during the period. The minimum, average, and maximum pipeline times are next displayed, followed by the standard deviation of the distribution. As with other reports, the reporting period and base number are initialized values.

INITIALIZATION

The NRTS Program requires the initialization of 16 variables. Only three, however, require values. The variable description and initialization table (Table 7) contains the information required to initialize the report program. An example data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

This is a supplement to the Bench Repair Analysis Program, having a very similar logic (only much simpler), and using the same input tape made up of 12-word label records. The output is a report (every period) on the passage of reparables through the NRTS cycle.

Only those records with IDD = 6000 or 4900 are of concern to this program; all others are skipped. As before, all records with base number not equal to the system attribute BASE are skipped

Because no ACCUMULATE statements are used, this program does not use the automatically defined system variable TIME, nor does it need a CLOCK subroutine.

An ERROR routine is present, and is called in case of an error condition from several places in the program.

NRTS DATA

BASE 1 PERIOD ENDING DAY 30.000

REP NO	QUANTITY NRTS	NRTS DELAY TIME MIN	AVG	MAX	STD DEV
1	3	10.566	13.658	17.940	3.126
2	12	5.018	9.181	14.283	2.578
3	10	2.086	2.529	3.669	0.471
5	5	5.967	8.536	11.769	2.224
6	1	3.136	3.138	3.138	0.000

Fig. 24

Table 7

VARIABLE DESCRIPTION AND INITIALIZATION:
NRTS

Array Number	Number of Subscripts	Mode		Initialize to		Initialize Value in		Array Number of Attribute to Be Entered in Fig. 5 Col.		List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
		Integer	Floating Point	Zero	Value	Table	Col.	19-22 (rows)	27-30 (cols.)					
1-4	0			X										
5	0	X			V						Base number.	BASE	S	
6	0		F		V						Report interval in decimal days.	PERIOD	S	
7	0	X			V						Total number of reparable types.	NPTYP	S	
8-16	1			X				7						A

PERMANENT VARIABLES

TIM - current simulated time, as read from the current label record.

PTIME - the time at the beginning of the current report period.

ENDPD - the time at which the current report period will end

CURPD - the length of the report period that has just ended.

BASE - the base number for this run.

PEROD - the length of a report period.

RPTYP - reparable type; a permanent entity with the following attributes:

FLOAD - first of the set called LOAD.

LLOAD - last of the set called LOAD.

TOTAL - the total number of reparables of this type completing the NRTS cycle in this period.

SUM - the sum of their NRTS times.

SUMSQ - the sum-square of their NRTS times.

AVG - the mean NRTS time for this type of reparable during this period.

STDV - the standard deviation of NRTS time for this type of reparable.

MAX - the maximum NRTS time for this type of reparable during this period.

TEMPORARY VARIABLES

REP - reparable part; a temporary entity with the following attributes:

IDNO - I.D. number; a number obtained from the label record that uniquely identifies this reparable.

BTIME - the time at which this reparable entered the NRTS cycle.

PLOAD - predecessor in LOAD.

SLOAD - successor in LOAD.

SETS

LOAD - a singly-subscripted set, ranked on BTIME.

owner: RPTYP

member: REP

```
*T REP  4
*                  T IDNO  1     I
*                  T BTIME 2     F
*                  T PLOAD 3     I
*                  T SLOAD 4     I
*                                                         LOAD1   *BTIME L
*                                   1TIM        F
*                                   2PTIME      F
*                                   3ENDPD      F
*                                   4CURPD      F
*                                   5BASE       IC
*                                   6PEROD      FC
*                                   7RPTYP E
*                                   8FLOAD 1    I
*                                   9LLOAD 1    I
*                                  10TOTAL 1    I
*                                  11SUM   1    F
*                                  12SUMSQ 1    F
*                                  13AVG   1    F
*                                  14STDV  1    F
*                                  15MAX   1    F
*                                  16MIN   1    F
*IBFTC MAIN
              MAIN ROUTINE
C      ......INITIALIZE ENDPD.
              LET ENDPD = PEROD
C      ......INITIALIZE EACH MINIMUM TO A HIGH NUMBER.
              LET MIN(I) = 10000., FOR EACH RPTYP I
C      ......READ A LABEL RECORD.
X  10         READ (9) K,IDD,K,K,K,INBASE,NO,K,K,IADDR,INDIC,T
              LET TIM = T
C      ......IF THERE IS A DETAIL RECORD, SKIP IT.
X             IF (INDIC.EQ.1) READ (9) JUNK
              IF (IDD) NE (3), GO TO 30
C      ......END OF SIMULATION. (IDD=3.) TERMINATE AFTER PRINTING THE
C      ......     LAST REPORT.
              LET ENDPD = TIM
              CALL ENDPRD
              STOP
C      ......IGNORE THIS RECORD UNLESS INBASE = BASE.
   30         IF (INBASE) NE (BASE), GO TO 10
C      ......TEST FOR END OF PERIOD.
              IF (TIM) GR (ENDPD), CALL ENDPRD
C      ......IF THIS RECORD IS RELEVANT (IDD = 6000 OR 4900), CALL A
C      ......     SUBROUTINE TO PROCESS IT.  OTHERWISE SKIP IT.
              IF (IDD) NE (6000), GO TO 50
              CALL START (NO, IADDR)
              GO TO 10
   50         IF (IDD) NE (4900), GO TO 10
              CALL FINISH (NO, IADDR)
              GO TO 10
              END
*IBFTC ENDPRD
              SUBROUTINE ENDPRD
```

```
C     ......THIS ROUTINE IS CALLED AT THE END OF EACH PERIOD.
            LET T = ENDPD
            LET CURPD = T - PTIME
            LET PTIME = T
C     ......OUTPUT NOTHING IF NO TIME HAS ELAPSED SINCE END OF PREVIOUS
C     ......     PERIOD.
            IF (CURPD) EQ (0.), GO TO 100
            DO TO 30, FOR EACH RPTYP I
            LET FTOT = TOTAL(I)
            CALL STDDEV (FTOT, SUM(I), SUMSQ(I), *AVG(I), *STDV(I))
            IF (MIN(I)) EQ (10000.), LET MIN(I) = 0.
   30       LOOP
            CALL REPORT
C     ......RESET ALL RUNNING TOTALS.
            DO TO 50, FOR EACH RPTYP I
            LET MIN(I) = 10000.
            LET MAX(I) = 0.
            LET SUM(I) = 0.
            LET SUMSQ(I) = 0.
            LET TOTAL(I) = 0
   50       LOOP
            LET ENDPD =  ENDPD + PEROD
  100       RETURN
            END
*IBFTC START
            SUBROUTINE START (NO, ID)
C     ......THIS ROUTINE IS CALLED FOR IDD = 6000.
C     ......A REP HAS JUST ENTERED THE NRTS CYCLE.
            IF (NO) GR (NRPTYP), CALL ERROR (4HSTAR)
            CREATE REP
            LET BTIME(REP) = TIM
            LET IDNO(REP) = ID
            FILE REP IN LOAD(NO)
            RETURN
            END
*IBFTC FINISH
            SUBROUTINE FINISH (NO, ID)
C     ......THIS ROUTINE IS CALLED FOR IDD = 4900.
C     ......A REP HAS JUST LEFT THE NRTS CYCLE.
            IF (NO) GR (NRPTYP), CALL ERROR (4HFINI)
C     ......FIND THE REP, TAKE STATISTICS, AND DESTROY IT.
            FIND FIRST, FOR EACH REP OF LOAD(NO), WITH (IDNO(REP)) EQ
     *        (ID), WHERE REP, IF NONE, CALL ERROR (4HFIN2)
            REMOVE REP FROM LOAD(NO)
            LET T = TIM - BTIME(REP)
            LET MAX(NO) = AMAX1(T,MAX(NO))
            LET MIN(NO) = AMIN1(T,MIN(NO))
            LET SUM(NO) = SUM(NO) + T
            LET SUMSQ(NO) = SUMSQ(NO) + T*T
            LET TOTAL(NO) = TOTAL(NO) + 1
            DESTROY REP
            RETURN
            END
*IBFTC STDDEV
```

```
            SUBROUTINE STDDEV (TOT, S, SC, AVE, STD)
C      ......ROUTINE TO TAKE A MEAN AND STANDARD DEVIATICN.
            IF (TOT) LE (0.), GO TO 50
            LET AVE = S / TOT
            LET STD = SQRT(AMAX1(SQ/TOT - AVE*AVE, 0.))
            GO TO 100
   50       LET AVE = 0.
            LET STD = 0.
  100       RETURN
            END
*IBFTC REPORT
      REPORT REPORT
X                                                                    NRTS
X                                                                     BASE
 X
X                                              REP     QUANTITY        NRTS
X                                              NO        NRTS        MIN
X                                              *          **         *.***
 X                                             I       TOTAL(I)      MIN(I)
  X   FOR EACH RPTYP I, WITH (TOTAL(I)) NE (0)
                                   END
                                                                        X
DATA                                                                1
  *   PERIOD ENDING DAY   **.***                                    1
BASE                      ENDPD
DELAY TIME
    AVG          MAX          STD DEV                               1
  **.***       **.***        **.***
  AVG(I)       MAX(I)        STDV(I)
                                                                        X
                                   END
*IBFTC ERROR
            SUBROUTINE ERROR (NAME)
            CALL ERREPT (NAME)
            STOP
            END
*IBFTC ERREPT
      REPORT ERREPT (NAME)
X     JOB TERMINATED AT TIME    **.*** BECAUSE OF ERROR IN SUBROUTINE '**
 X                                 TIM                                 NA
X                                                                  (ABBRE
                                   END
                                                                        X
A*'
ME
VIATION)
                                   END
```

```
1          16
   1   4    Z
   5        R                                            1          BASE
   6        R                                            5.0        PEROD
   7        R                                            20         NRPTYP
   8  16 1  Z    20    7
```

Program 11

DEPOT TRANSPORTATION CAPABILITY

XI. DEPOT TRANSPORTATION CAPABILITY

The Depot Transportation Capability Output Program is a two-part program displaying the cargo delivered to each base(s) and the utilization of each transport vehicle.

CARGO

The Cargo Report (see Fig. 25) displays the quantity of cargo moved throughout the simulated period. The quantity is specified in terms of weight, volume, and units for each type of cargo. Note the separate specification of both the quantity of cargo that is loaded aboa some carrier (TP) and the quantity of cargo delivered (TD) by a carrier. The same distinction is made for base deliveries (column 4) and for the depot (column 5). Column 6 specifies the quantity of each cargo type that is in process (in transit) as of the report time.

INITIALIZATION

The cargo output program requires the user to initialize eighteen variables. Table 8 and its accompanying text will facilitate use of this program, and help the user understand the conceptual basis for the resultant report.

OUTPUT PROGRAM

The input to the cargo output program is the tape generated by the Depot Transportation Simulation Program. The tape is read from logical unit No. 9.

This input tape consists of twelve-word records in the following form:

Word 1 - irrelevant.

Word 2 - a four-digit number identifying the occurrence represented by this record.

Words 3 through 5 - irrelevant.

Word 6 - the base number.

CARGO REPORT FOR BASE 2 PERIOD ENDING 4.00

CARGO TYPE	TONS DELIVERED TP	TONS DELIVERED TD	UNITS DELIVERED TP	UNITS DELIVERED TD	VOLUME DELIVERED TP	VOLUME DELIVERED TD	BASE UNITS TP	BASE UNITS TD	DEPOT UNITS TP	DEPOT UNITS TD	IN PROCESS
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0

Fig. 25

Table 8

VARIABLE DESCRIPTION AND INITIALIZATION:
CARGO

Array Number	Number of Subscripts	Mode: Integer	Mode: [illegible]	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col.: 19-22 (pens)	Array Number of Attribute to Be Entered in Fig. 5 Col.: 27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1	0		R								Report Interval	CIR	X	
2	0	I									No. of bases contained in the simulation.	BASE	X	
3	0	I									No. of reparables contained in the simulation.	NRPS	X	
4	2	I		X				3	2			TTOR		A
5	2	I		X				3	2			TWOL		A
6	2	I		X				3	2			TMIR		A
7	2	I		X				3	2			TOPQ		A
8	2	I		X				3	2			TRAG		A
9	0		R	X								LIRT		A
10	1	I			X			3			Specify the weight of each reparable contained in the simulation.	WRT		A
11	2	I						3	2			FRPQ		A
12	2	I						3	2			LRPQ		A
13	1	I			X			3			Specify the volume of each reparable.	VOL		A
14	0	I			X						First base number to be printed by this report.	FBAST		A
15	0	I			X						Last base number to be printed by this report.	LBAST		A
16	1	I			X			2			Desired sequence of base numbers to be printed	BBAST		A
17	1	I		X				2				FBASP		A
18	1	I		X				2				LBASP		A

Words 7 through 10 - the content of these words varies with the individual values of Word 2.

Word 11 - irrelevant.

Word 12 - current simulated time.

When a record is read from tape 9, the value of CNT is compared with current simulated time (Word 12); if simulated time equals or exceeds the value of CNT, steps are taken to issue a report. Thus, CNT is used to control the report interval. If CNT exceeds the value of simulated time, the current record's information is processed as follows.

Word 6 is checked to determine if the record contains information about a relevant base. There is provision in the cargo output program for reporting on all, or selected, bases, in any order desired by the user.

For all relevant bases, Word 2 is compared with internal four-digit codes in order to select only records that are "interesting" to the program. At this point, the program transfers control to segments unique to each code, which will in turn process the individual data records.

When Word 2 is found to equal 5000, the end of simulation has been reached. A final report is issued, and the cargo output program terminates.

PERMANENT VARIABLES

The following list is complete except for attributes denoting first- or last-of-set.

CNT - the report interval, specified by the user.

BASE - a permanent entity representing the number of bases in the simulation.

NRPS - the total number of unique reparables involved in the simulation.

TTØN - sum of the tonnage arriving at each base.

TVØL - sum of the volume arriving at each base.

TUNT - sum of the number of units arriving at each base.

TDPØ - sum of the number of base units arriving at each base (base unit = reparable that originates at a base rather than a depot).

TBAS - sum of the number of depot units arriving at each base.

LNT - a debugging aid.

WHT - a list of weights for individual reparable types.

VØL - a list of volumes for individual reparable types

TEMPORARY VARIABLES

REP - a temporary entity with the following attributes:

BFLG - a system flag indicating that the reparable is a base unit.

SYSTA - the reparable type number (as opposed to serial number, which is a machine address).

WATE - weight of the particular reparable.

VOLU - volume of the particular reparable.

RPORT - a temporary entity with the following attributes:

IUNIT - a running sum of units delivered to each base. It is reset to zero after each report interval.

TØTØN - a running sum of total weight delivered to each base. It is reset to zero after each report interval.

TØVØL - a similar running sum for total volume.

DPØU - a similar running sum for total depot units.

BASU - a similar running sum for total base units.

INPRØ - a running sum of "in-process" reparables (total units - (sum of depot and base units)).

SETS

BREP - a singly-subscripted FIFØ set consisting of all temporary REPs attached to each base.

Owner: BASE

Member: REP

RPQ - a doubly-subscripted FIFØ set consisting of all temporary RPØRTs filed by base and reparable type.

Owners: NRPS, BASE

Member: RPØRT

BSET - a FIFØ set, with no subscripts, which contains the numbers of those bases for which a report is to be issued.

Owner: SYSTEM

Member: BASE

```
+                                         1CNT    0     FC
+                                         2BASE   E     I
+                                         3NRPS   0     IC
+                                         4TTON   2     I
+                                         5TVOL   2     I
+                                         6TUNT   2     I
+                                         7TDPC   2     I
+                                         8TBAS   2     I
+                                         9LNT    0     FC
+                                        10WHT    1     I
+                                        11FRPC   2     I
+                                        12LRPC   2     I
<                                        13VOL    1     I
+                                        14FBSET  0     I
+                                        15LBSET  0     I
+                                        16SBSET  1     I
+                                        17FBREP  1     I
+                                        18LPREP  1     I
+T REP   8
+                    T RELG   11/2 I
+                    T SYSTA  21/2 I
+                    T WATE   22/2 I
+                    T VOLC   31/2 I
+                    T SBREP  51/2 I
+
+T RPORT8                                                 RREPI *
+                    T TUNIT  11/2 I
+                    T TOTON  12/2 I
+                    T TOVOL  21/2 I
+                    T PPOU   22/2 I
+                    T BASU   31/2 I
+                    T SRPO   52/2 I
+                    T INPRO  61/2 I
+
+                                                         RPG 2 *
*IBFTC MAIN                                               PSETO *
      MAIN
C
C.....MAIN PREPARES CARGO PARAMETERS.
C
      LET GNT = CNT
    1 READ FROM 9, ID,IDS,KOD,KAY,KPV,IB,ILA,ILB,ILC,ITR,IDT,TYME
      FORMAT(5I6,S6,I3,S6,3I6,S6,I6,S6,I2,S6,D3.5)
      IF(TYME)LS(GNT), GO TO 200
      CALL REPORT(GNT)
  100 LET GNT = GNT + CNT
C
```

```
C.....NOT REPORT TIME, PROCESS RECORD.
C
  200 FIND FIRST, FOR EACH K IN BSET, WITH (K) EQ(IB), IF NONE, GO TO 1
      IF (IDS) EQ (5000),GO TO 500
      IF (IDS) EQ (5060),GO TO 550
      IF (IDS) EQ (507 ),GO TO 550
      IF (IDS) EQ (5500),GO TO 550
      GO TO 1
C
C.....IDS = 5000, END OF SIMULATION.
C
  500 CALL REPORT(GNT)
      STOP
C
  550 CREATE REP
      STORE  WHT(ILB) IN WATE(REP)
      STORE  VOL(ILB) IN VOLU(REP)
      STORE  ILB IN SYSTA(REP)
      IF (IDS) NE (5060),  LET BFLG(REP) = 1
      FILE REP IN BREP(IB)
      GO TO 1
      END
*IBFIC BLOK2
      SUBROUTINE REPORT(GNT)
C
C.....COLLECT PARAMETERS.
C
C.....INCREMENT 'THIS PERIOD' DATA.
C
  101 DO TO 10, FOR EACH IB IN BSET
      DO TO  9, FOR J = (1)(NRPS)
      CREATE RPORT CALLED LIST
    1 DO TO 3, FOR EACH REP IN BREP(IB), WITH (SYSTA(REP)) EQ (J)
      LET IUNIT(LIST) = IUNIT(LIST) + 1
      LET TOTON(LIST) = TOTON(LIST) + WATE(REP)/2000
      LET TOVOL(LIST) = TOVOL(LIST) + VOLU(REP)
      IF(BFLG(REP))EQ(1), GO TO 2
      LET DPOU(LIST) = DPOU(LIST) + 1
      GO TO 3
    2 LET BASU(LIST) = BASU(LIST) + 1
    3 REPEAT 1
C
C.....COMPUTE 'IN PROCESS' TOTAL FOR THIS REPORT PERIOD.
C
      LET KPROC = IUNIT(LIST) - (BASU(LIST) + DPOU(LIST))
      STORE KPROC IN INPRO(LIST)
C
C.....FILE REPORT IN QUEUE OF REPORTS BY BASE AND SYSTEM.
C
      FILE LIST IN RPO(J,IB)
C
C.....INCREMENT 'TO DATE' COUNTERS.
C
```

```
      LET TTON(J,IB) = TTON(J,IB) + TOTON(LIST)
      LET TVOL(J,IB) = TVOL(J,IB) + TOVOL(LIST)
      LET TUNT(J,IB) = TUNT(J,IB) + TUNIT(LIST)
      LET TDPO(J,IB) = TDPO(J,IB) +  DPOU(LIST)
      LET TBAS(J,IB) = TBAS(J,IB) +  BASU(LIST)
C
C.....PROCEED TO REPS  AT NEXT BASE.QUEUE.
C
    9 LOOP
C
C.....PROCURE NEXT BASE.
C
   10 REPEAT 101
C
C.....THERE IS NOW A REPORT FOR EACH REP. BY BASE.
C
C
C.....CALL REPORT GENERATOR PRELUDE.
C
      CALL FORM(GNT)
C
C.....HOUSEKEEP BEFORE RETURNING TO SUPERVISOR.
C
  301 DO TO 30, FOR EACH IB IN BSET
C
   22 IF BREP(IB) IS EMPTY, GO TO 23
      REMOVE FIRST J FROM BREP(IB)
      DESTROY REP CALLED J
      GO TO 22
   23 DO TO 24, FOR J = (1)(NRPS)
      IF RPQ(J,IB) IS EMPTY, GO TO 24
      REMOVE FIRST RPORT FROM RPQ(J,IB)
      DESTROY RPORT
   24 LOOP
   30 REPEAT 301
      RETURN
      END
*IBFTC BLOK3
      SUBROUTINE FORM(GNT)
C
C.....PREPARE TO PRINT HEADING.
C
      LET KNT = 0
      LET MARK= 0
    1 DO TO 6, FOR EACH IB IN BSET
    2 CALL FORMH(IB,GNT)
      LET KNT = KNT + 10
      IF(MARK)EQ(1), GO TO 4
C
C.....PRINT DATA RECORD.
C
      DO TO 5, FOR K = (1)(NRPS)
    3 DO TO 4, FOR EACH J IN RPQ(K,IB)
```

```
      CALL FORMR(K,TOTON(J),TTON(K,IB),IUNIT(J),TUNT(K,IB),TOVOL(J),
     1TVOL(K,IB),DPOU(J),TDPO(K,IB),BASU(J),TBAS(K,IB),INPRO(J))
      LET MARK = 0
      LET KNT  = KNT + 1
      IF (KNT) EQ (55), GO TO 7
    4 REPEAT 3
    5 LOOP
    6 REPEAT 1
      RETURN
    7 LET KNT = C
      LET MARK= 1
      GO TO 2
      END
*IBFTC FORMH
      REPORT FORMH(IB,GNT)
X                                                   CARGO  REPORT  FOR  BASE
 X
X                                          TONS                 UNITS
X                       CARGO          DELIVERED             DELIVERED
X                       TYPE         TP          TD        TP          TC
      END
                                                                              X
*  PERIOD  ENDING    *.**                                                9
IB                   GNT
   VOLUME                      BASE            DEPOT
  DELIVERED                    UNITS           UNITS         IN
TP          TD            TP        TD     TP       TD     PROCESS          2
END
*IPFTC FORMR
      REPORT FORMR(IA,IB,IC,ID,IE,IF,IG,L,M,N,IC,IP)
X                      *          *          *          *          *
 X                     IA         IB         IC         ID         IE
      END
                                                                              X
  *         *         *       *       *       *       *                     X
IF          IG        L       M       N       IO      IP
END
*ENTRY            MAIN
1         18
   1      0 R                                            0.5
   2      0 R                                            10
   3      0 R                                            17
   4    8 2  Z     17    3  10    2
   9      0  Z
  10      1 R     17    3                                12(16)
011386 10386  8386  7768  8768  9768  7573  8573  9573 14449 14449   200
000200 14685 10462 10462 10462
  11   12 2  7     17    3  10    2
  13      1 R     17    3                                17(14)
0110 11C  7C 370 370 370 140 140 140 100 180  10  10 110  40  90  40
  14      0 R                                            2
  15      0 R                                            5
  16      1 R     10    2                                10(12)
 0 4 5 3
  17   18 1  7     10    2
```

UTIL

The UTIL (utilization) program describes the Cargo Carrier Utilization for the simulation. The utilization of each vehicle (by ID number) for each vehicle type is listed. Figure 26 is an example of the information contained in this portion of the report.

For each vehicle, col. 3 lists the time the vehicle was available for service, col. 4 the time lost due to maintenance, and col. 5 the time involved in loading the vehicle. Idle time, listed in col. 6, is the sum of maintenance downtime and loading time. The utilization factor is the sum of maintenance time, loading time, and intransit time, divided by the total simulated time to date.

INITIALIZATION

The UTIL output program requires the user to initialize nine variables. Table 9 and its accompanying text will facilitate the use of the program and help the user understand the conceptual basis for the resultant report.

OUTPUT PROGRAM

The input to UTIL is the tape generated by the Depot Transportation Simulation program. The tape is read from logical unit No. 9.

This input tape consists of twelve-word records in the following format:

Word 1 - irrelevant.

Word 2 - a four-digit number identifying the occurrence represented by this record.

Words 3 through 5 - irrelevant.

Word 6 - the base number.

Words 7 through 10 - the content of these words varies with the individual values of Word 2.

Word 11 - irrelevant.

Word 12 - current simulated time.

When a record is read from tape 9, the value of GNT is compared with current simulated time (Word 12); if simulated time equals or

UTILIZATION OF VEHICLE TYPE 4 FOR PERIOD ENDING 3.00

VEHICLF TYPF	VEHICLE ID	TIME AVAILABLE	MAINTENANCE DOWNTIME	LOADING TIME	IDLE TIME	UTILIZATION FACTOR
4	23951	1.00	0.00	0.00	1.00	0.00
4	23943	1.00	0.00	0.00	1.00	0.00
4	23935	1.00	0.00	0.00	1.00	0.00

Fig. 26

Table 9

VARIABLE DESCRIPTION AND INITIALIZATION:
UTILIZATION

Array Number	Number of Subscripts	Mode		Initialize to		Initialize Value in		Array Number of Attribute to Be Entered in Fig. 5 Col.		List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
		Integer	Floating Point	Zero	Value	Table	Col.	19-22 (rows)	27-30 (cols.)					
1	0		F		V						Report Interval	CNY	E	
2	0	I			V						Number of cargo carrier types	NCARS	E	
3	1	I		Z				2				FBUSQ		A
4	1	I		Z				2				LBUSQ		A
5	1	I		Z				2				FLSTQ		A
6	1	I		Z				2				LISTQ		A
7	0	I									First vehicle number to be reported - by type	FVSET		A
8	0	I									Last vehicle number to be reported - by type	LVSET		A
9	1	I						2			Desired sequence of reports or selected vehicle types	SVSET		A

exceeds the value of GNT, steps are taken to issue a report. Thus, GNT is used to control the report interval. If GNT exceeds the value of simulated time, the current record's information is processed as follows.

Word 2 is compared with internal four-digit codes in order to select, for further processing, only records that are relevant to this program's objective. Once such a record is identified, program control is transferred to one of several unique segments corresponding to individual four-digit codes.

When Word 2 is found to equal 5000, the end of simulation has been reached. A final report is issued, and UTIL terminates.

PERMANENT VARIABLES

CNT - the report interval, specified by the user.

NCARS - total number of vehicles in the simulation.

FBUSQ - machine address representation of first member of the set BUSQ.

LBUSQ - the last member of BUSQ.

FLSTQ - machine address representation of first member of the set LSTQ.

LLSTQ - the last member of LSTQ.

FVSET, LVSET - first and last members of the set VSET.

SVSET - a list of successive members of VSET, starting with the successor to member FVSET.

TEMPORARY VARIABLES

BUS - a temporary entity with the following attributes:

TYPE - an integer specifying the particular kind of vehicle.

SRNØ - a machine address identifying an individual member of any one TYPE.

MAJØR - a running sum, for this vehicle, of major maintenance elapsed time.

MINØR - a running sum, for this vehicle, of minor maintenance elapsed time.

FLITE - a running sum of total travel time accrued by this vehicle.

LØAD - a running sum of time taken to load this vehicle.

T501 - T509 - these represent the simulated times relative to unique events associated with the vehicle during the simulation. They are used to compute the four preceding attributes.

LIST - a temporary entity with the following attributes:

MNTDT - total downtime due to maintenance for this vehicle, this report period.

LØDTM - total time spent in loading this vehicle during this period.

IDLE - vehicle idle time during this report period.

UTL - utility factor for an individual vehicle during this report period.

THISL - machine ID of this particular report.

SETS

BUSQ - a singly-subscripted FIFØ set consisting of all temporary entities called BUS (the vehicles).

Owner: SYSTEM

Member: BUS

LSTQ - a singly-subscripted FIFØ set consisting of all temporary entities called LIST (the individual reports).

Owner: SYSTEM

Member: LIST

```
*                                            1CNT     0      FC
*                                            2NCARS 0       I
*                                            3FBUSQ 1       I
*                                            4LBUSQ 1       I
*                                            5FLSTQ 1       I
*                                            6LLSTQ 1       I
*                                            7FVSET 0       I
*                                            8LVSET 0       I
*                                            9SVSET 1       I
*T BUS   88
*                   T SBUSQ 21/2 I
*                   T TYPE  22/2 I
*                   T SRNO  31/2 I
*                   T MAJOR 4    F
*                   T MINOR 5    F
*                   T FLITE 6    F
*                   T LOAD  7    F
*                   T T501  8    F
*                   T T502 11    F
*                   T T504 12    F
*                   T T509 13    F
*                   T FL50214    I
*                   T FUGE 15    F
*T LIST 8                                                  BUSQ1 *
*                   T MNTDT 2    F
*                   T LODTM 3    F
*                   T IDLE  4    F
*                   T UTL   5    F
*                   T SLSTQ 6    I
*                   T THISL 7    I
*IBFTC MAIN                                                LSTQ1 *
      MAIN
C
C.....UTILITY FACTOR OUTPUT PROGRAM. SET - UP.
      LET GNT = CNT
    1 READ FROM 9, ID,IDS,KOD,KAY,KPW,IB,ILA,ILB,ILC,ITR,IDT,TYMF
      FORMAT(5I6,S6,I3,S6,3I6,S6,I6,S6,I2,S6,D3.5)
      IF(TYMF)LS(GNT), GO TO 2
      CALL UPREP(GNT)
      LET GNT = GNT + CNT
C
    2 IF (IDS) EQ (5000), GO TO 5
      IF (IDS) EQ (5010), GO TO 10
      IF (IDS) EQ (5020), GO TO 20
      IF (IDS) EQ (5030), GO TO 30
      IF (IDS) EQ (5040), GO TO 40
      IF (IDS) EQ (5050), GO TO 50
```

```
      IF (IDS) EQ (5090), GO TO 60
      IF (IDS) EQ (5200), GO TO 70
      IF (IDS) EQ (5300), GO TO 80
      GO TO 1
C
C.....ORDER PARAMETERS.
C
   80 CREATE BUS CALLED K
      STORE ILA IN TYPE(K)
      STORE ITR IN SRNO(K)
      FILE K IN BUSQ(ILA)
      GO TO 1
   70 FIND FIRST, FOR EACH K IN BUSQ(ILA), WITH (SRNC(K))EQ(ITR), IF
     1NONE, GO TO 1
      IF(T509(K))LS(GNT-CNT),LET FUGE(K)=FUGE(K)+((GNT-CNT)-T509(K))
      LET MAJOR(K) = MAJOR(K) + (TYME - T509(K))
      GO TO 1
C
   60 DO TO 59, FOR J = (1)(NCARS)
      FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNC(K))EQ(ITR), IF NONE,
     1GO TO 59
      STORE TYME IN T5C9(K)
      GO TO 1
   59 LOOP
      GO TO 1
   5C DO TO 49, FOR J =(1)(NCARS)
      FIND FIRST, FOR EACH K IN BUSQ(J), WITH(SRNO(K))EQ (ITR), IF NONE,
     1GO TO 49
      IF(T504(K))LS(GNT-CNT),LET FUGE(K)=FUGE(K)+((GNT-CNT)-T504(K))
      LET FLITE(K) = FLITE(K) + (TYME - T504(K))
      GO TO 1
   49 LOOP
      GO TO 1
   40 DO TO 39, FOR J = (1)(NCARS)
      FIND FIRST,FOR EACH K IN BUSQ(J), WITH (SRNC(K)) EQ(ITR), IF NONE,
     1GO TO 39
      STORE TYME IN T504(K)
      IF(T502(K))EQ(0.), GO TO 41
      IF(T5C2(K))LS(GNT-CNT),LET FUGE(K)=FUGE(K)+((GNT-CNT)-T502(K))
      LET LOAD(K)=LOAD(K)+(T504(K)-T502(K))
      GO TO 42
   41 IF(T501(K))LS(GNT-CNT),LET FUGE(K)=FUGE(K)+((GNT-CNT)-T501(K))
      LET LOAD(K) = 0.
   42 IF (T502(K))EQ(0.), LET MINOR(K) = MINOR(K) + (T504(K) - T501(K))
      IF(T502(K)) NE (0.), LET T502(K) = 0.
      GO TO 1
   39 LOOP
      GO TO 1
   30 GO TO 20
   2C DO TO 19, FOR J = (1)(NCARS)
      FIND FIRST, FOR EACH K IN BUSC(J), WITH (SRNC(K))EC(ITR), IF NONE,
     1GO TO 19
      IF(T5C1(K))LS(GNT-CNT),LET FUGE(K)=FUGE(K)+((GNT-CNT)-T501(K))
```

```
      LET MINOR(K) = MINOR(K) + (TYME - T501(K))
      STORE TYME IN T502(K)
      GO TO 1
   19 LOOP
      GO TO 1
   10 DO TO 9, FOR J = (1)(NCARS)
      FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNO(K))EQ(ITR), IF NONE,
     1GO TO 9
      STORE TYME IN T501(K)
      GO TO 1
    9 LOOP
      GO TO 1
    5 CALL UPREP(GNT)
      STOP
      END
*IBFTC BLOK5
      SUBROUTINE UPREP(GNT)
C
C.....COLLECT PARAMETERS.
C
    1 DO TO  4, FOR EACH IV IN VSET
    2 DO TO 3, FOR EACH K IN BUSQ(IV)
      CREATE LIST CALLED L
      LET THISL(L) = SRNO(K)
      LET MNTOT(L) = MNTOT(L) + (MAJOR(K) + MINOR(K))
      LET LODTM(L) = LODTM(L) + LOAD(K)
      LET IDLE(L)=IDLE(L)+(CNT-((FLITE(K)+MNTOT(L)+LODTM(L))-FUGE(K)))
      LET UTL(L)   = UTL(L)   + (CNT-IDLE(L))/ CNT
      FILE L IN LSTQ(IV)
    3 REPEAT 2
    4 REPEAT 1
      CALL FORM(GNT)
C
C.....HOUSEKEEP BEFORE RETURNING TO SUPERVISOR.
C
    5 DO TO 8, FOR EACH IV IN VSET
   61 DO TO 6, FOR EACH BUS IN BUSQ(IV)
      LET MAJOR(BUS) = 0.
      LET MINOR(BUS) = 0.
      LET FLITE(BUS) = 0.
      LET  LOAD(BUS) = 0.
      LET FUGE(BUS) = 0.
    6 REPEAT 61
C
    7 IF LSTQ(IV) IS EMPTY, GO TO 8
      REMOVE FIRST LIST FROM LSTQ(IV)
      DESTROY LIST
      GO TO 7
    8 REPEAT 5
      RETURN
      END
*IBFTC BLOK6
      SUBROUTINE FORM(GNT)
```

```
C
C.....PREPARF TO PRINT HEADING.
C
      LET KNT = 0
      LET MARK= 0
    1 DO TO  5, FOR EACH IV IN VSET
    2 CALL FORMH(IV,GNT)
      LET KNT = KNT + 10
      IF (MARK)EQ(1),GO TO 4
C
C.....PRINT DATA RECORD.
C
    3 DO TO  4, FOR EACH L IN LSTQ(IV)
      IF (UTL(L)) EQ(-0.), LET UTL(L) = 0.
      LET AVL = CNT - MNTDT(L)
      CALL FORMR(IV,THISL(L),AVL,MNTDT(L),LODTM(L),IDLE(L),UTL(L))
      LET MARK = 0
      LET KNT = KNT + 1
      IF (KNT) FQ (55), GO TO 6
    4 REPEAT 3
    5 REPEAT 1
      RETURN
    6 LET KNT = 0
      LET MARK= 1
      GO TO 2
      END
*IBFTC FORMH
      REPORT     FORMH(IV,GNT)
X                                                 UTILIZATION  OF  VEHICLE  TYPE
 X
X                                  VEHICLE     VEHICLE      TIME          MAINTE
X                                   TYPF         ID       AVAILABLE        DOWNT
      END
                                                                              X
   *    FOR  PERIOD  ENDING   *.**                                           9
  IV                           GNT
NANCE       LOADING        IDLE       UTILIZATION
IME           TIME         TIME         FACTOR                               2
END
*IBFTC FURMR
      REPORT     FORMR(IV,L,A1,A2,A3,A4,A5)
X                                  *            *          *.**             *.
 X                                IV            L           A1              A2
      END
                                                                              X
**             *.**          *.**          *.**                             X
                A3            A4            A5
END
*ENTRY            MAIN
1           9
   1      0 R                                          1.0
   2      0 R                                          4
   3    6 1  7      4    2                             1
   7      0 R                                          1
   8      C R                                          4
   9      1 R       4    2                             4(I2)
 2 3 4
```

Program 12

DEPOT MAINTENANCE CAPABILITY

XII. DEPOT MAINTENANCE CAPABILITY

The Depot Maintenance Capability program is used to display the outputs from the DR&O Simulator.[4] The report consists of five parts: the input to each depot, its output and the reparable repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of simulation.

An example of the output display is shown in Figs. 27 to 31. Figure 27 is a display of depot statistics showing the system (or Unit or item) arrivals and departures, reparables in process, and repair cycle times. A separate display is presented for each depot. The example display is for depot No. 3.

The first line of data shows the time at which the statistics were taken. Notice that the report is for day <u>ending</u> 14.000. Since "time" began at time 0.000 in the simulations and the report is initialized for seven-day periods, the fourteenth day will end at time 14.000 (not 14.999). The next line entry shows the number of reparables that entered the depot (15) for the period and the sum of all reparables entering the depot (30) as of the report period.

The third line entry is the serviceables departing the depot (returned to serviceable stock) (8), during the period, and the sum of all items processed to date (20). The difference between the arrivals and departures is presented as the number in process (10).

The next line displays a distribution of the reparables in process. The average time in process (for the twenty that were processed) is 4.75 days, the maximum time was 10 days, and the minimum 1 day. The standard deviation for the distribution is 2.37.

The repair time distribution is presented both for the period (just 7 days) and accumulated for the fourteen days.

Figure 28 displays the activity Queueing Factors for the period. Column 1 lists the activities in sequence. Columns 2, 3, and 4 list the

distribution of the quantity of reparables processed by each activity for the period. Column 5 lists the average time that the reparables spend in queue behind each activity awaiting some resource. The average queue time is displayed as work time; i.e., off-shift time is not included.

Figure 29 is the Personnel Utilization report. For each personnel type, listed in col. 1, the sum of all personnel on duty for all shifts of the period (of course, the period may be only one shift) is presented in col. 2. The utilization factor, which is the time actually engaged in a process divided by the total duty time available, is presented in col. 3 for each personnel type. The balance of the display is devoted to the man-hours used at each activity for each personnel type. For example, personnel type 2 worked at activities 5, 10, and 11, and a total of 93.25 man-hours were used during the simulation period (seven days).

Figure 30 is the Equipment Utilization report. By equipment type, listed in col. 1, the quantity is listed in col. 2; cols. 3, 4, and 5 list the time the equipment was used, the idle time, and the downtime (all in decimal-days). Note that the summation of these three columns is equal to 14 equipment days for Equipment types 1 and 2, and 21 equipment days for Equipment type 3. This is the total time available for the equipment. Off-shift time is not deducted.

Column 6 is a count of the number of times the depot equipment failed during the period (in this example, 7 days). Column 7 is the utilization factor for the equipment, computed by dividing the total time available (equipment days) into the time in use. Column 8 lists the activities where the equipment was used.

Figure 31 is the display of the Queueing factors for each component spare part. Column 1 lists the spare part ID number. Column 2 lists the quantity or authorized stock level of each spare part. Column 3 lists the number of demands for each spare part during the period.

Columns 4, 5, and 6 list the distribution of the quantity of unfilled demands (average, maximum, and minimum) for each spare part type. Column 7 lists the average queue time--the average time required to fill the demand.

DEPOT 3

SYSTEM ARRIVALS, DEPARTURES, IN-PROCESS,
AND REPAIR CYCLE TIMES

DATA FOR PERIOD ENDING DAY 14.000

REPARABLES ENTERING DEPOT THIS PERIOD 15, TO DATE 30

SERVICEABLES DEPARTING DEPOT THIS PERIOD 8, TO DATE 20

NUMBER IN PROCESS 10

	AVG	MAX	MIN	STD DEV
REPARABLES IN PROCESS	4.75	10	1	2.37
REPAIR TIME				
THIS PERIOD	1.99	3.03	0.96	0.69
TO DATE	1.37	3.03	0.88	0.67

Fig. 27

ACTIVITY QUEUEING FACTORS
FOR PERIOD JUST COMPLETED

ACT. NO.	NO. OF REPS IN QUEUE AVG	MAX	MIN	AVG QUEUE TIME (IN WORK-HOURS)
1	0.33	3.	0.	0.80
2	0.05	3.	0.	0.63
3	0.00	1.	0.	0.00
4	0.05	1.	0.	0.09
5	1.10	5.	0.	0.76
6	0.01	1.	0.	0.02
7	3.18	8.	0.	1.58
8	0.00	1.	0.	0.00
9	4.15	10.	0.	7.44
10	0.27	3.	0.	0.26
11	0.00	1.	0.	0.00
12	0.00	2.	0.	0.53
13	0.01	1.	0.	0.01

Fig. 28

PERSONNEL UTILIZATION

PERS TYPE	QTY	UTIL FACT	NO	MAN-HOURS	NO	WORK TIME AT ACTIVITY MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS	NO	MAN-HOURS
1	42	0.09	1	4.41	2	11.82	3	4.83	12	3.01	13	5.79		
2	36	0.32	5	20.40	10	64.10	11	8.75						
3	48	0.27	4	32.03	5	40.79	10	32.05						
4	30	0.40	6	32.22	7	24.18	8	22.85	9	15.63				

Fig. 29

EQUIPMENT UTILIZATION

EQUIPMENT TYPE	QUANTITY	TIME IN USE	IDLE TIME	DOWN TIME	NO. OF FAILURES	UTILIZATION FACTOR	ACTIVITY NOS. WHERE USED
1	2	0.67	13.15	0.18	3	0.048	4
2	2	1.26	12.35	0.39	11	0.090	10
3	3	1.70	19.29	0.01	1	0.081	5

Fig. 30

QUEUEING FACTORS BY COMPONENT SPARES TYPE

REPARABLE QUEUE LENGTHS AND TIMES BY COMPONENTS

COMPONENT TYPE	QUANTITY	DEMANDS THIS PERIOD	NO. OF REPS IN QUEUE AVG	MAX	MIN	AVG QUEUE TIME (IN WORK-HOURS)
1	10	26	0.00	1.	0.	0.00
2	10	20	0.86	3.	0.	0.00
3	8	21	4.30	9.	0.	0.00
4	12	18	0.00	1.	0.	0.00
5	11	27	12.79	21.	6.	0.00

Fig. 31

INITIALIZATION

The Depot Capability report program requires the initialization of 83 variables. Only 12 require values, however. The Depot Capability Variable Description and Initialization Table (Table 10) contains the information required to initialize the report program. An example initialization data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

The input to this program is the binary tape generated by the DR&O Simulation Program; this tape is read from logical unit No. 9.

The input tape consists of 12-word label records with the following format:

Word 1 - irrelevant.

Word 2 - IDD - a four-digit number identifying the "event" or "occurrence" represented by this record.

Word 3, 4, 5 - irrelevant.

Word 6 - INBASE - the depot number.

Words 7, 8, 9, 10 - IV1, IV2, IV3, LADDR. These fields are used to store various items of information, depending on the value of IDD.

Word 11 - INDIC - 1 if the next record is a detail record (to be skipped), 0 otherwise.

Word 12 - RTIME - current simulated time.

When a label record is read, the value of INBASE is compared with the constant permanent attribute called BASE; if they are unequal, the record is skipped. (Thus it would require n runs of this analysis program to process all the data from an n-base simulation run, each time changing the value of BASE).

If the new RTIME is greater than the previous one, subroutine CLOCK is called to check for the end of the operating shift and the end of the report period. If the report period has ended, subroutine ENDPRD is called to generate the reports. Subroutine CLOCK also updates TIME, which is the actual work time elapsed since the beginning of

Table 10

VARIABLE DESCRIPTION AND INITIALIZATION: DEPOT MAINTENANCE CAPABILITY

Array Number	Number of Subscripts	Mode: Integer	Mode: Floating Point	Initialize to: Zero	Initialize to: Value	Initialize Value in: Table	Initialize Value in: Col.	Array Number of Attribute to Be Entered in Fig. 5 Col.: 19-22 (rows)	27-30 (cols.)	List Packing	Description of Variable to Be Initialized	Permanent System Variable Name	Entity	Attribute
1-9	0			Z										A
10	0	I			V						Total number of depots.	SHOP	E	
11-29	1			Z				10						A
30	0	I			V						Total number of Activities.	ACTIV	E	
31-39	1			Z				30						A
40	0	I			V						Total number of Personnel Types.	PTYPE	E	
41-45	1			Z				40						A
46	0	I			V						Total number of Equipment Types.	ETYPE	E	
47-48	1			Z				46						A
49	0			Z										A
50-57	1			Z				46						A
58-59	0			Z										A
60	0	I			V						Total number of Spare Part Types.	SPTYP	E	
61-71	1			Z				60						A
72-75	0			Z										A
76	0	I			V						Total number of Reparable Types.	RPTYP	E	
77	1	I			V			76			List depot number where each reparable type is processed. (one data card/rep type).	SHPRO		A
78	0	I			V						Total number of shifts per week.	SHIFT	E	
79	1	I			V			78			Insert a "1" for each shift worked and a "0" for each shift not worked (one data card/shift).	SCHED		A
80	2	I		Z				40	78		Insert quantity of Personnel type "P" at shift "N" (one data card/personnel type).	QTYPR		A
81	0		F		V						Total number of hours per shift.	LENGH		A
82	0		F		V						Report interval in decimal days.	PRIOD		A
83	0	I			V						Base number	BASE		A

simulation. (The automatically defined system variable TIME is used, in order to take advantage of the ACCUMULATE statement.)

Then the appropriate subroutine is called to process the label record. To each significant IDD number, there corresponds a subroutine: e.g., subroutine NEXTAC is called whenever IDD equals 4400. If IDD does not match any of the significant numbers, it is skipped.

If IDD = 3, the end of simulation has been reached; the program terminates after writing the last set of reports.

Error tests intended for the debugging phase have been left in the program, sprinkled throughout. If an error is encountered, this means that something is amiss in this program, in the simulation program, or in the initialization deck. Subroutine ERROR is called, which terminates after outputting the current value of RTIME and a four-letter abbreviation identifying the routine in which the error was detected. For instance, "REA2" refers to the second error condition in subroutine READY.

Subroutine SNAP outputs a "snapshot" of all permanent and temporary variables, as an aid to debugging. The user may insert, at any point, a call to SNAP with an identifier of one to four letters and/or digits; e.g., CALL SNAP (4HNAME). In this example, "NAME" is the identifier. In the current version of the program, ERROR calls SNAP before terminating.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set or last-of-set, and attributes used only to keep track of time in an ACCUMULATE statement such as TQSZA (these always have names beginning with "T").

RTIME - current simulated time; it is obtained from each label record as it is read in.

STIME - the "RTIME" of the previous label record.

TIME - (a variable automatically defined by the system) - number of <u>workdays</u> elapsed since the beginning of simulation. Suppose there are 40 work-hours in a week. Then if RTIME = 7.0, TIME will be equal to 1.6667 or 1-2/3 (which is 40 divided by 24).

PTIME - the value of "TIME" at the end of the previous report period.

ENDSH - the "RTIME" at which the current shift will end.

ENDPD - the "RTIME" at which the current period will end.

ETIME - the value of "RTIME" at the end of the previous report period.

CURPD - the length in workdays (using "TIME") of the period just completed.

CURP - the length in simulated time (using "RTIME") of the period just completed.

CURSH - number of current shift (on a weekly cycle).

CURAC - activity number associated with current label record.

CURSP - spare part number associated with current label record.

CUREP - I.D. number of REP associated with current label record.

SHOP - permanent entity, of which the following are attributes:

RIN - number of reps entering this depot this period.

ROUT - number of reps leaving depot this period.

TRIN - total number of reps in depot (since the beginning of simulation).

TROUT - total number of reps that have left this depot.

MAXR - maximum number of reps in depot this period.

MINR - minimum number of reps this depot this period.

RIP - number of reps currently in process in this depot.

RIPS - a running sum of all the values that RIP has assumed during this period.

RIPSQ - a running sum-square total of all the values that RIP has assumed during this period. E.g., if RIP has had the values 2, 3, 4, 3, 2 in this period, then RIPS is $2 + 3 + 4 + 3 + 2$ or 14, and RIPSQ is $2^2 + 3^2 + 4^2 + 3^2 + 2^2$ or 42.

RTS - sum of the repair times of all reps leaving depot this period.

RTSQ - sum of squares of repair times of all reps leaving depot this period.

TRTS - sum of RTS for all periods to date.

TRTSQ - sum of RTSQ for all periods to date.

MXRT - maximum repair time for depot this period.

MNRT - minimum repair time for depot this period.

TMXRT - maximum repair time for depot, <u>all</u> periods.

TMNRT - minimum repair time for depot, <u>all</u> periods.

ACTIV - activity; a permanent entity, of which the following are attributes:

QSZA - current queue size at this activity.

CQSZA - cumulative total of QSZA, this period.

MXQSA - minimum value of QSZA, this period.

MNQSA - minimum value of QSZA, this period.

TIAQS - "time in activity queue, summed;" the total time, in workdays, that reps have spent in the queue for this activity.

AVQSA - average queue size at this activity.

AVTAQ - average time in queue for this activity.

AQOUT - number of reps that have left the queue of this activity during this period.

PTYPE - personnel type; a permanent entity, of which the following are attributes:

QTYS - total number of this type of personnel.

CQTY - number of man-days for this personnel type for this period.

ETYPE - equipment type; a permanent entity with the following attributes:

QTYE - total quantity of this equipment type.

NFAIL - number of failures of this type of equipment during this period.

INUSE - quantity of this equipment type currently in use.

CINUS - cumulative total of INUSE, this period.

DOWN - quantity of this type of equipment that is currently down.

CDOWN - cumulative total of DOWN, this period.

SPTYP - spare part type; a permanent entity with the following attributes:

QTYSP - quantity of spares of this type available at beginning of simulation.

DMAND - number of demands for this type of part during this period.

FILL - number of times that such a demand was filled.

QSP - queue size for this type of part.

CQSP - cumulative total of QSP, this period.

MXQSP - maximum value of QSP, this period.

MNQSP - minimum value of QSP, this period.

TISQS - total time, in work days, that reps have spent in the queue for this type of part.

AVQSP - average value of QSP, this period.

AVTSQ - average time in queue for this type of spare part.

RPTYP - rep type; a permanent entity with the following attribute:

SHPNO - number of the depot to which this type of rep belongs.

SHIFT - a permanent entity with the following attribute:

SCHED - 1 if this is a work shift; 0 if this is an off shift.

QTYPR - a permanent attribute with two subscripts:

first subscript: PTYPE

second subscript: SHIFT

meaning: the quantity of personnel of this type, on duty during this shift.

LENSH - the length of a shift.

PEROD - the length of a report period.

BASE - the number of the depot for this run; all label records pertaining to any other depot will be ignored.

TEMPORARY VARIABLES

REP - a temporary entity with the following attributes:

QTIME - the value of "TIME" when the rep entered the queue for an activity.

ETIME - the value of "RTIME" when the rep entered the system.

IDNO - the I.D. number of the rep: a number obtained from the label record, representing the absolute storage address of the rep in the simulation run.

QFLAG - a number which is equal to zero unless the rep is in the queue for an activity, in which case QFLAG equals the number of that activity.

SFLAG - equal to zero unless rep is in the queue for a spare part, in which case SFLAG equals the number of that type of part.

SPTIM - the value of "TIME" when the rep entered the queue for a spare part.

PLOAD, SLOAD, PACTQ, SACTQ - attributes associated with the sets LOAD and ACTQ.

DUMMY - a temporary entity whose purpose is to save information to be output in Fig. 28. It has two attributes:

ACNO - the number of an activity at which this type of equipment is to be used.

SSET - successor in the set called "SET."

ENTRY - a temporary entity having to do with the utilization of personnel at different activities. Its attributes are:

ACNO - the number of an activity at which this type of personnel is used.

WKING - number of personnel of this type working at activity whose number equals ACNO.

CWKNG - cumulative total of WKING, this period.

TWKNG - the value of "RTIME" when CWKNG was last updated.

PLIST, SLIST - attributes associated with the set called "LIST."

SETS

LOAD - a set with one subscript, ranked on BTIME.

Owner: SHOP

Member: REP

The LOAD of each SHOP consists of all the reps that are currently in process in that shop.

ACTQ - a set with one subscript, ranked on BTIME.

Owner: ACTIV

Member: REP

ACTQ is the queue of all reps currently waiting at an activity.

SET - a FIFO set with one subscript.

Owner: ETYPE

Member: DUMMY

SET is the set of all activities at which this type of equipment can be used. This information is to be output in Fig. 30.

LIST - a set with one subscript, ranked on ACNO.

Owner: PTYPE

Member: ENTRY

LIST has one ENTRY for each activity at which this type of personnel has been used during this report period.

Standard Names for Local Variables

IACNO	always means activity number
IEQNO	always means equipment number
IPERNO	always means personnel number
ISPNO	always means spare part number
IREPNO	always means rep number
ID or IDREP	always means I.D. number of rep
IQTY	always means quantity or number

```
**********            DEPOT CAPABILITY CUTPUT PROGRAM                                ****
*T REP   8
*                  T CTIME 1     F
*                  T BTIME 2     F
*                  T IDNC  3     I
*                  T GFLAG 41/2  I
*                  T SFLAG 42/2  I
*                  T SPTIM 5     F
*                  T SLOAD 6     I
*                  T SACTQ 7     I
*                  T PLOAD 81/2  I
*                  T PACTQ 82/2  I
*                                                        LOAD1   *RTIME L
*                                                        ACTQ1   *HTIME L
*T DUMMY2
*                  * SSET  2     I
*                                                        SET 1 *
*T ENTRY8
*                  T ACNO  1     I
*                  T WKING 2     F
*                  T CWKNG 3     F
*                  T TWKNG 4     F
*                  T PLIST 5     I
*                  T SLIST 6     I
*                                                        LIST1   *ACNO  L
*                              1RTIME        F
*                              2STIME        F
*                              3PTIME        F
*                              4ENCSH        F
*                              5ENCPC        F
*                              6CURPC        F
*                              7CURSH        I
*                              8CURAC        I
*                              9CURSP        I
*                             10SHCP   E
*                             11FLCAD  1     I
*                             12LLCAC  1     I
*                             13RIN    1     I
*                             14RCUT   1     I
*                             15TRIN   1     I
*                             16TRCUT  1     I
*                             17MAXR   1     I
*                             18MINR   1     I
*                             19RIP    1     I
*                             20RIPS   1     F
*                             21RIPSC  1     F
*                             22RTS    1     F
*                             23RTSC   1     F
*                             24TRTS   1     F
*                             25TRTSC  1     F
*                             26MXRT   1     F
*                             27MNRT   1     F
*                             28TMXRT  1     F
```

```
29TMNRT 1    F
30ACTIV E
31CSZA  1    F
32CCSZA 1    F
33TCSZA 1    F
34MXCSA 1    F
35MNCSA 1    F
36TIACS 1    F
32AVCSA 1    F
36AVTAQ 1    F
37FACTC 1    I
38LACTC 1    I
39ACCUT 1    I
40PTYPE E
41FLIST 1    I
42LLIST 1    I
43CTYS  1    I
44CCTY  1    F
45TCTY  1    F
46ETYPE E
47CTYE  1    I
48NFAIL 1    I
50INUSE 1    F
51CINUS 1    F
52TINUS 1    F
53DOWN  1    F
54CDOWN 1    F
55TDOWN 1    F
56FSET  1    I
57LSET  1    I
60SPTYP E
61CTYSP 1    I
62DMAND 1    I
63FILL  1    I
64FSPC  1    I
65LSPC  1    I
66CSP   1    F
67CCSP  1    F
68TQSP  1    F
69MXCSP 1    F
70MNQSP 1    F
71TISCS 1    F
67AVQSP 1    F
71AVTSC 1    F
73CUREP      I
74CURP       F
75ETIME      F
76RPTYP E
77SHPNC 1     IC
78SHIFT E
79SCHED 1     IC
80CTYPR 2    F
81LENSH      FC
82PEROD      FC
83BASE        IC
```

```
*IBFTC MAIN
              MAIN ROUTINE
              CALL PRELIM
C     ......READ A LABEL RECORD
X   1C        READ (9) K,IDD,K,K,K,INBASE,IV1,IV2,IV3,IADDR,INDIC,T
              LET RTIME = T
C     ......IF THERE IS A DETAIL RECORD, SKIP OVER IT
X             IF (INDIC.EQ.1) READ (9) JUNK
C     ......TERMINATE IF AN ENDSIM RECORD (WITH IDD=3) IS ENCOUNTERED
              IF (IDD) NE (3), GO TO 30
              CALL CLOCK
              LET ENDPD = RTIME
              CALL ENDPRD
              STOP
C     ......SKIP THIS RECORD IF IT DOES NOT PERTAIN TO THE RIGHT BASE
   3C         IF (INBASE) NE (BASE), GC TC 10
              IF (RTIME) GR (STIME), CALL CLOCK
C     ......CALL THE APPROPRIATE ROUTINE FOR THIS IDD NUMBER
              IF (IDD) EQ (7000), GO TC 50
              IF (IDD) EQ (7002), GC TC 52
              IF (IDD) EQ (7003), GC TC 54
              IF (IDD) EQ (7004), GO TC 56
              IF (IDD) EQ (7005), GC TC 58
              IF (IDD) EQ (7200), GC TC 60
              IF (IDD) EQ (6000), GC TC 62
              IF (IDD) EQ (7400), GO TC 64
              IF (IDD) EQ (7401), GO TC 66
              IF (IDD) EQ (7460), GC TC 68
              IF (IDD) EQ (7550), GC TC 70
              IF (IDD) EQ (7370), GO TC 72
              IF (IDD) EQ (7560), GO TC 74
              IF (IDD) EQ (7600), GC TC 76
              IF (IDD) EQ (7700), GC TC 78
              IF (IDD) EQ (7350), GO TC 80
              IF (IDD) EQ (7355), GC TC 82
              IF (IDD) EQ (7800), GC TC 84
              IF (IDD) EQ (7801), GC TC 86
C     ......FOR ANY OTHER VALUE OF IDD, SKIP THIS RECORD
              GO TO 10
   5C         CALL ACTVTY (IV1)
              GO TC 1C
   52         CALL EQATAC (IV1)
              GC TC 10
   54         CALL PRSNEL (IV1, IV2, IV3)
              GO TO 1C
   56         CALL EQUIP (IV1, IV2)
              GO TO 1C
   58         CALL SPARES (IV1, IV2)
              GO TO 1C
   60         CALL ARRIV (IV1, IADDR)
              GO TO 10
   62         CALL DEPART (IV1, IADDR)
              GO TO 1C
```

```
64        CALL NEXTAC (IV1, IV3, IACDR)
          GO TO 1C
66        CALL READY (IV3)
          GO TO 1C
68        CALL ASINPR (IV1, IV3)
          GO TO 1C
7C        CALL RLESPR (IV1, IV3, IADDR)
          GO TO 1C
72        CALL ASINEC (IV1, IV3)
          GO TO 1C
74        CALL RLESEQ (IV1, IV3)
          GO TO 10
76        CALL FAIL (IV3)
          GO TO 10
78        CALL RESTOR (IV3)
          GO TO 10
8C        CALL SPAVL (IV1)
          GO TO 1C
82        CALL INSPQ (IV1, IV2)
          GO TO 1C
84        CALL SPRET (IV1)
          GO TO 10
86        CALL LVSPQ (IV1, IADDR)
          GO TO 10
          END


*IBFTC PRELIM
          SUBROUTINE PRELIM
C     ......INITIALIZE SCPE SYSTEM VARIABLES
          LET ENDPD = PERCD
          LET ENDSH = LENSH
          LET CURSH = 1
C     ......INITIALIZE EACH MINIMUM TC A VERY LARGE NUMBER
          DO TC 20, FOR EACH SHOP I
          LET MNRT(I) = 1000C.
          LET TMNRT(I) = 1C000.
   2C     LOOP
          RETURN
          END
```

```
*IBFTC CLOCK
            SUBROUTINE CLOCK
C      ......THIS ROUTINE KEEPS TRACK OF TIME, END-OF-PERIOD, AND END-OF-
C      ......     SHIFT.  'TIME' IS THE ACTUAL WORK-TIME ELAPSED SINCE
C      ......     THE BEGINNING OF SIMULATION, WHEREAS 'RTIME' IS THE
C      ......     CURRENT SIMULATED TIME.
   20       LET T = AMIN1 (RTIME, ENDSH, ENDPD)
C      ......UPDATE TIME IF SOME WORK-TIME HAS ELAPSED, THAT IS, IF THIS
C      ......     IS A WORKING SHIFT
            IF (SCHED(CURSH)) EQ (1), LET TIME = TIME + T - STIME
C      ......UPDATE STIME
            LET STIME = T
            IF (RTIME) EQ (T), GO TO 100
C      ......UPDATE THE NO. OF MAN-DAYS (CQTY) FOR EACH PERSONNEL TYPE I
            DO TO 40, FOR EACH PTYPE I
            ACCUMULATE QTYPR(I,CURSH) INTO CQTY(I) SINCE TQTY(I)
   40       LOOP
C      ......THERE IS AN END-OF-SHIFT AND/OR AN END-OF-PERIOD. DETERMINE
C      ......     WHICH CAME FIRST.
            IF (ENDSH) LE (ENDPD), GO TO 50
C      ......END-OF-PERIOD
            CALL ENDPRD
            GO TO 20
C      ......END-OF-SHIFT. UPDATE CURSH AND ENDSH.
   50       LET CURSH = MOD (CURSH, NSHIFT) + 1
            LET ENDSH = ENDSH + LENSH
            GO TO 20
  100       RETURN
            END


*IBFTC ENDPRD
            SUBROUTINE ENDPRD
C      ......END OF A REPORT PERIOD.
C      ......COMPUTE CURP AND CURPD.
            LET S = STIME
            LET CURP = S - ETIME
            LET ETIME = S
            LET CURPD = TIME - PTIME
            LET PTIME = TIME
C      ......IF NO WORK-TIME HAS ELAPSED, DON'T OUTPUT ANYTHING
            IF (CURPD) EQ (0.), GO TO 100
C      ......OUTPUT THE REPORTS FOR THIS PERIOD
            CALL OUT1
            CALL OUT2
            CALL OUT3
            CALL OUT4
            CALL OUT5
            LET ENDPD = ENDPD + PERCD
  100       RETURN
            END
```

```
*IBFTC ACTVTY
          SUBROUTINE ACTVTY (IACNC)
C     ......THIS RCUTINE IS CALLED WHEN ICC=7000. (AT BEGINNING OF RUN)
          IF (IACNO) GR (NACTIV), CALL ERROR (4HACTV)
          LET CURAC = IACNC
          RETURN
          END
```

```
*IBFTC EQATAC
          SUBROUTINE EQATAC (IEGNC)
C     ......THIS ROUTINE IS CALLED WHEN ICC=7002. (AT BEGINNING OF RUN)
          IF (IEQNO) GR (NETYPE), CALL ERROR (4HEQAT)
C     ......SAVE ACTIVITY NUMBERS FCR REPCRT NC. 4
          CREATE DUMMY CALLED ITEM
          LET ACNO(ITEM) = CURAC
          FILE ITEM IN SET(IECNO)
          RETURN
          END
```

```
*IBFTC PRSNEL
          SUBROUTINE PRSNEL (IPERNC, ICTY, ISHIFT)
C     ......THIS ROUTINE IS CALLED WHEN ICC=7003. (AT BEGINNING OF RUN)
          IF (IPERNO) GR (NPTYPE), CALL ERRCR (4HPRSN)
          LET QTYS(IPERNO) = QTYS(IPERNC) + ICTY
          LET QTYPR(IPERNC,ISHIFT) = ICTY
          RETURN
          END
```

```
*IBFTC EQUIP
          SUBROUTINE EQUIP (IEGNC, ICTY)
C     ......THIS ROUTINE IS CALLED WHEN IDC=7004. (AT BEGINNING OF RUN)
          IF (IEGNO) GR (NETYPE), CALL ERRCR (4HEQUI)
          LET QTYE(IEQNC) = ICTY
          RETURN
          END
```

```
*IBFTC SPARES
          SUBROUTINE SPARES (ISPNC, IQTY)
C      ......THIS ROUTINE IS CALLED WHEN ICD=7005. (AT BEGINNING OF RUN)
          IF (ISPNC) GR (NSPTYP), CALL ERRCR (4HSPAR)
          LET QTYSP(ISPNC) = IQTY
          RETURN
          END
```

```
*IBFTC ARRIV
          SUBROUTINE ARRIV (IREPNC, ID)
C      ......THIS ROUTINE IS CALLED WHEN ICC=7200.
C      ......A REP HAS ENTERED THE SYSTEM. CREATE A TEMPORARY RECORD FOR
C      ......     IT, FILE IT INTO THE APPROPRIATE DEPOT, AND UPDATE THE
C      ......     STATISTICS FOR THIS DEPOT.
          CREATE REP
          LET BTIME(REP) = RTIME
          LET IDNO(REP) = ID
          LET ISHOP = SHPNO(IREPNC)
          LET NEWRIP = RIP(ISHOP) + 1
          LET RIP(ISHOP) = NEWRIP
          LET RIN(ISHOP) = RIN(ISHOP) + 1
          LET MAXR(ISHOP) = MAXO (NEWRIP, MAXR(ISHOP))
          LET FRIP = NEWRIP
          LET RIPS(ISHOP) = RIPS(ISHOP) + FRIP
          LET RIPSQ(ISHOP) = RIPSQ(ISHOP) + FRIP**2
          FILE REP IN LOAD(ISHOP)
          RETURN
          END
```

```
*IBFTC DEPART
          SUBROUTINE DEPART (IREPNC, IC)
C      ......THIS ROUTINE IS CALLED WHEN ICC=6000.
C      ......A REP HAS LEFT THE SYSTEM. REMOVE AND DESTROY IT.
          LET ISHOP = SHPNO(IREPNC)
          FIND FIRST REP, FOR EACH REP IN LOAD(ISHOP), WITH
     *      (IDNO(REP))EQ(ID), WHERE REP, IF NONE, CALL ERROR (4HDEPA
          REMOVE REP FROM LOAD(ISHOP)
          LET NEWRIP = RIP(ISHOP) - 1
          LET RIP(ISHOP) = NEWRIP
          LET ROUT(ISHOP) = ROUT(ISHOP) + 1
          LET MINR(ISHOP) = MINO (NEWRIP, MINR(ISHOP))
          LET FRIP = NEWRIP
          LET RIPS(ISHOP) = RIPS(ISHOP) + FRIP
          LET RIPSQ(ISHOP) = RIPSQ(ISHOP) + FRIP**2
          LET REPTIM = RTIME - BTIME(REP)
          LET MXRT(ISHOP) = AMAX1 (REPTIM, MXRT(ISHOP))
          LET MNRT(ISHOP) = AMIN1 (REPTIM, MNRT(ISHOP))
          LET RTS(ISHOP) =  RTS(ISHOP) + REPTIM
          LET RTSQ(ISHOP) = RTSQ(ISHOP) + REPTIM**2
          DESTROY REP
          RETURN
          END


*IBFTC NEXTAC
          SUBROUTINE NEXTAC (IREPNC, IACNC, IDREP)
C      ......THIS ROUTINE IS CALLED WHEN ICC=7400.
C      ......A REP IS SUBMITTED (OR RE-SUBMITTED) TO THIS ACTIVITY.
          LET CURAC = IACNC
C      ......DO NOTHING IF ACTIVITY = 0 (RECEIVING).
          IF (IACNO) EQ (0), GO TO 50
          LET CUREP = IDREP
C      ......FIND THE REP BY SEARCHING THE LOAD OF THE APPROPRIATE DEPOT
          FIND FIRST REP, FOR EACH REP IN LOAD(SHPNO(IREPNO)), WITH
     *      (IDNO(REP)) EQ (IDREP), WHERE REP, IF NONE, CALL
     *      ERROR (4HNEXT)
          LET K = QFLAG(REP)
C      ......IF QFLAG EQUALS THE ACT. NO., THIS REP IS ALREADY IN THE
C      ......        QUEUE FOR THIS ACTIVITY, SO DO NOTHING.
          IF (K) EQ (IACNC), GO TO 50
C      ......IF REP IS ALREADY IN SOME OTHER QUEUE, THIS IS AN ERROR.
          IF (K) NE (0), CALL ERROR (4HNEX2)
C      ......IF QFLAG = 0, FILE IT INTO THE QUEUE FOR THIS ACTIVITY.
          LET QFLAG(REP) = IACNC
          ACC QSZA(IACNO) INTO CQSZA(IACNC) SINCE TQSZA(IACNO), ADD 1
          LET MXQSA(IACNO) = AMAX1 (QSZA(IACNC), MXQSA(IACNO))
          LET QTIME(REP) = TIME
          FILE REP IN ACTQ(IACNC)
   50     RETURN
          END
```

```
*IBFTC ASINPR
          SUBROUTINE ASINPR (IPERNC, IQTY)
C     ......THIS ROUTINE IS CALLED WHEN IOC=7460.
C     ......ASSIGN PERSONNEL TO AN ACTIVITY.
          LET Q = IQTY
          LET R = RTIME
          LET IACNO = CURAC
          IF (IACNO) EQ (C), CALL ERRCR (4HASPR)
C     ......IF THIS IS THE FIRST TIME (DURING THIS REPORT PERIOD) THAT
C     ......      PERSONNEL OF THIS TYPE HAVE BEEN ASSIGNED TO THIS
C     ......      ACTIVITY, CREATE A NEW ENTRY AND FILE IT INTO LIST FOR
C     ......      THIS PERSONNEL NO. IN ANY CASE, TAKE STATISTICS.
          FIND FIRST, FOR EACH ENTRY OF LIST(IPERNC), WITH
     *      (ACNO(ENTRY)) EQ (IACNO), WHERE ENTRY, IF NONE, GO TO 20
          LET W = WKING(ENTRY)
          LET CWKNG(ENTRY) = CWKNG(ENTRY) + W * (R-TWKNG(ENTRY))
          LET WKING(ENTRY) = W + Q
          GO TO 50
   20     CREATE ENTRY
          LET ACNO(ENTRY) = IACNO
          LET WKING(ENTRY) = Q
          FILE ENTRY IN LIST(IPERNC)
   50     LET TWKNG(ENTRY) = R
          RETURN
          END


*IBFTC READY
          SUBROUTINE READY (IACNO)
C     ......THIS ROUTINE IS CALLED WHEN IOC=7401.
C     ......THIS REP IS READY TO BE WORKED ON. REMOVE IT FROM QUEUE FOR
C     ......      THIS ACTIVITY, AND TAKE STATISTICS.
          ACC QSZA(IACNO) INTO CQSZA(IACNO) SINCE TQSZA(IACNO),ACC -1
          IF (QSZA(IACNO)) LE (-1.), CALL ERRCR (4HREA1)
          FIND FIRST, FOR EACH REP IN ACTQ(IACNO), WITH (IONO(REP)) E
     *      (CUREP), WHERE REP, IF NONE, CALL ERRCR (4HREA2)
          REMOVE REP FROM ACTQ(IACNO)
          LET MNQSA(IACNO) = AMIN1 (QSZA(IACNO), MNQSA(IACNO))
          LET TIACS(IACNO) = TIACS(IACNO) + TIME - QTIME(REP)
          LET AQOUT(IACNO) = AQOUT(IACNO) + 1
          LET QFLAG(REP) = 0
          RETURN
          END
```

```
*IBFTC RLESPR
              SUBROUTINE RLESPR (IPERNC, ICTY, IACNC)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7550.
C       ......PERSONNEL HAVE BEEN RELEASED FROM THIS ACTIVITY. TAKE
C       ......     STATISTICS.
              FIND FIRST, FOR EACH ENTRY OF LIST(IPERNC), WITH
     *          (ACNC(ENTRY)) EQ (IACNC), WHERE ENTRY, IF NONE, CALL
     *          ERROR (4HRLPR)
              LET W = WKING(ENTRY)
              LET R = RTIME
              LET CWKNG(ENTRY) = CWKNG(ENTRY) + W * (R-TWKNG(ENTRY))
              LET TWKNG(ENTRY) = R
              LET WKING(ENTRY) = W - FLCAT(ICTY)
              RETURN
              END


*IBFTC ASINEQ
              SUBROUTINE ASINEQ (IEQNC, ICTY)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7370.
C       ......ASSIGN EQUIPMENT.
              ACCUMULATE INUSE(IEQNC) INTO CINUS(IEQNC) SINCE
     *          TINUS(IEQNC), ADD FLCAT(ICTY)
              RETURN
              END


*IBFTC RLESEQ
              SUBROUTINE RLESEQ (IEQNC, ICTY)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7560.
C       ......RELEASE EQUIPMENT.
              ACCUMULATE INUSE(IEQNC) INTO CINUS(IEQNC) SINCE
     *          TINUS(IEQNC), ADD -FLCAT(ICTY)
              IF (INUSE(IEQNC)) LE (-1.), CALL ERROR (4HRLEQ)
              RETURN
              END
```

```
*IBFTC FAIL
            SUBROUTINE FAIL (IEQNO)
C       ......THIS ROUTINE IS CALLED WHEN ICC=7600.
C       ......EQUIPMENT FAILURE.
            LET NFAIL(IEQNO) = NFAIL(IEQNO) + 1
            LET T = RTIME
            LET CDOWN(IEQNO) = CDOWN(IEQNO) + DOWN(IEQNO) *
     *                                          (T - TDOWN(IEQNO))
            LET TDOWN(IEQNO) = T
            LET DOWN(IEQNO) = DOWN(IEQNO) + 1.
            RETURN
            END


*IBFTC RESTOR
            SUBROUTINE RESTOR (IEQNO)
C       ......THIS ROUTINE IS CALLED WHEN ICC=7700.
C       ......EQUIPMENT RESTORED.
            LET T = RTIME
            LET CDOWN(IEQNO) = CDOWN(IEQNO) + DOWN(IEQNO) *
     *                                          (T - TDOWN(IEQNO))
            LET TDOWN(IEQNO) = T
            LET DOWN(IEQNO) = DOWN(IEQNO) - 1.
            IF (DOWN(IEQNO)) LE (-1.), CALL ERROR (4HREST)
            RETURN
            END


*IBFTC SPAVL
            SUBROUTINE SPAVL (ISPNO)
C       ......THIS ROUTINE IS CALLED WHEN ICC=7350.
C       ......THERE IS A DEMAND FOR A SPARE. THE SPARE IS AVAILABLE, SO
C       ......      THE DEMAND IS IMMEDIATELY FILLED.
            LET DMAND(ISPNO) = DMAND(ISPNO) + 1
            LET FILL(ISPNO) = FILL(ISPNO) + 1
            LET MXQSP(ISPNO) = AMAX1 (QSP(ISPNO)+1., MXQSP(ISPNO))
            RETURN
            END
```

```
*IBFTC INSPC
              SUBROUTINE INSPC (ISPNO, IREPNO)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7355.
C       ......THERE IS A DEMAND FOR AN UNAVAILABLE SPARE PART.
              FIND FIRST, FOR EACH REP IN LCAD(SHPNO(IREPNO)), WITH
     *          (IDNO(REP)) EQ (CUREP), WHERE REP, IF NONE, CALL
     *          ERROR (4HINSP)
              LET S = SFLAG(REP)
C       ......IF SFLAG = SPARE PART NO., THIS REP IS ALREADY IN QUEUE FOR
C       ......      THIS PART, SO DO NOTHING.
              IF (S) EQ (ISPNO), GO TO 50
C       ......IF REP IS ALREADY IN QUEUE FOR A DIFFERENT PART, CALL ERROR
              IF (S) NE (0), CALL ERROR (4HINS2)
C       ......IF SFLAG = 0, PUT IT IN QUEUE FOR THIS SPARE PART NO.
              LET SFLAG(REP) = ISPNO
              LET DMAND(ISPNO) = DMAND(ISPNO) + 1
              ACC QSP(ISPNO) INTO CQSP(ISPNO) SINCE TQSP(ISPNO), ADD 1.
              LET MXQSP(ISPNO) = AMAX1 (QSP(ISPNO), MXQSP(ISPNO))
              LET SPTIM(REP) = TIME
   50         RETURN
              END


*IBFTC SPRET
              SUBROUTINE SPRET (ISPNO)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7800.
C       ......(ALL WE NEED FROM THIS LABEL RECORD IS THE SPARE PART NO.)
              LET CURSP = ISPNO
              RETURN
              END


*IBFTC LVSPQ
              SUBROUTINE LVSPQ (IREPNO, IDREP)
C       ......THIS ROUTINE IS CALLED WHEN IDC=7801.
C       ......A SPARE PART IS AVAILABLE.
C       ......IF NO REP WAS WAITING FOR THIS PART, DO NOTHING.
              IF (IDREP) EQ (0), GO TO 50
C       ......TAKE THIS REP OUT OF THE QUEUE FOR THIS SPARE PART.
              LET ISPNO = CURSP
              LET FILL(ISPNO) = FILL(ISPNO) + 1
              ACC QSP(ISPNO) INTO CQSP(ISPNO) SINCE TQSP(ISPNO), ADD -1.
              IF (QSP(ISPNO)) LE (-1.), CALL ERROR (4HLVSP)
              LET MNQSP(ISPNO) = AMIN1 (QSP(ISPNO), MNQSP(ISPNO))
              FIND FIRST, FOR EACH REP IN LLAD(SHPNO(IREPNO)), WITH
     *          (IDNO(REP)) EQ (IDREP), WHERE REP, IF NONE, CALL
     *          ERROR (4HLVS2)
              LET TISQS(ISPNO) = TISQS(ISPNO) + TIME - SPTIM(REP)
              LET SFLAG(REP) = 0
   50         RETURN
              END
```

```
*IBFTC OUT1
          SUBROLTINE OLT1
C     ......GENERATE A 'TABLE 1' REPCRT FCR EACH CEPOT.
          DO TO 1CC, FOR EACH SHCP I
          LET IRIN = RIN(I)
          LET IRCUT = ROUT(I)
          LET FROLT = IROLT
          LET FRTS = RTS(I)
          LET FRTSC = RTSC(I)
          LET IRIP = RIP(I)
          LET FRIP = IRIP
          LET ITRIN = TRIN(I) + IRIN
          LET TRIN(I) = ITRIN
          LET ITROLT = TRCUT(I) + IRCUT
          LET TRCLT(I) = ITRCUT
          LET FTRCLT = ITRCUT
          LET TOTAL = IRIN + IROUT + 1
          CALL STDDEV (TOTAL, RIPS(I), RIPSC(I), *AVN, *STCVN)
          IF (IROLT) EQ (0), LET MART(I) = 0.
          CALL STDDEV (FRCUT, FRTS, FRTSC, *AVT, *STDVT)
          LET FTRTS = TRTS(I) + FRTS
          LET TRTS(I) = FTRTS
          LET FTRTSQ = TRTSC(I) + FRTSC
          LET TRTSQ(I) = FTRTSC
          CALL STDDEV (FTROUT, FTRTS, FTRTSQ, *TAVT, *TSTDVT)
          IF (ITROLT) EQ (C), GO TC 60
          LET TMXRT(I) = AMAX1 (MXRT(I), TMXRT(I))
          LET FTMNRT = AMIN1 (MNRT(I), TMNRT(I))
          LET TMNRT(I) = FTMNRT
          GO TO 8C
   6C     LET FTMNRT = 0.
   8C     CALL TAB1 (I, AVN, STDVN, AVT, STCVT, TAVT, FTMNRT, TSTDVT)
C     ......RESET VARIABLES FOR NEXT REPCRT PERICC.
          LET RIN(I) = C
          LET ROLT(I) = 0
          LET MAXR(I) = IRIP
          LET MINR(I) = IRIP
          LET RIPS(I) = FRIP
          LET RIPSQ(I) = FRIP * FRIP
          LET RTS(I) = 0.
          LET RTSC(I) = 0.
          LET MXRT(I) = 0.
          LET MNRT(I) = 10C00.
  1CC     LOOP
          RETURN
          END
```

```
*IBFTC STDDEV
           SUBROUTINE STDDEV (TCTAL, SUM, SUMSC, AVG, STCV)
C     ......ROUTINE TO CCMPUTE A MEAN ANC STANCARC CEVIATION.
           IF (TOTAL) LE (C.), GO TC 50
           LET AVG = SUM / TCTAL
           LET STDV = SQRT(AMAX1(SUMSC/TCTAL - AVC*AVG, 0.))
           GO TO 100
   5C      LET AVG = C.
           LET STDV = 0.
  1CC      RETURN
           END
```

```
*IBFTC TAB1
      REPORT TAB1 (I, AVN, STDVN, AVT, STCVT, TAVT, FTMNRT, TSTDVT)
X                                                                       CEP
 X
X                                                         SYSTEM ARRIVALS, DE
X                                                                 ANC REPAIR
X
 X
X                                              REPARABLES ENTERING CEPOT THIS
 X
X                                             SERVICEABLES CEPARTING DFPOT THI
 X
X
 X
X                                                                      AVG
X                                              REPARABLES IN PRCCESS    **.**
 X                                                                     AVN
X                                              REPAIR TIME
X                                                 THIS PERIOD           *.**
 X                                                                     AVT
X                                                 TC DATE               *.**
 X                                                                     TAVT
                              END

T   *                                                                   1
   I
ARTURES, IN-PROCESS,
CYCLE TIMES                                                             1
DATA FOR PERIOD ENDING DAY  **.***                                      1
                            ENDPD
 'IOD        **, TO DATE  ***                                           1
          RIN(I)         TRIN(I)
 PERIOD      **, TO DATE  ***                                           1
          ROUT(I)        TROUT(I)
  NUMBER IN PROCESS        **                                           1
                         RIP(I)
   MAX       MIN     STD DEV                                            1
   **         *        *.**                                             1
  MAXR(I)   MINR(I)  STDVN
                                                                        1
  **.**      *.**      *.**                                             1
  MXRT(I)   MNRT(I)  STDVT
  **.**      *.**      *.**
 TMXRT(I)   FTMNRT  TSTDVT
                                 END
```

```
*IBFTC OUT2
              SUBROUTINE OUT2
C      ......GENERATE TABLE 2.
              LET C = CURPD
              DO TO 5C, FOR EACH ACTIV I
              ACCUMULATE QSZA(I) INTO CQSZA(I) SINCE TQSZA(I)
              LET AVQSA(I) = CQSZA(I) / C
              IF (AQOUT(I)) EQ (0), GO TO 50
              LET AVTAQ(I) = DECHR(TIACS(I)) / FLOAT(AQOUT(I))
   5C         LOOP
              CALL TAB2
              DO TO 1CC, FOR EACH ACTIV I
              LET CQSZA(I) = 0.
              LET TIACS(I) = 0.
              LET FQSZA = QSZA(I)
              LET MXQSA(I) = FQSZA
              LET MNQSA(I) = FQSZA
              LET AQOUT(I) = 0
  1CC         LOOP
              RETURN
              END


*IBFTC TAB2
       REPORT TAB2
X                                                                          ACTIVITY QU
X                                                                          FOR PERIOD .
X                                                                          NO. OF REPS
X                                                            ACT.
X                                                             NC.            AVG    MAX
X                                                             **            *.**    **.
 X                                                             I        AVQSA(I) MXQSA
  X   FOR EACH ACTIV I, WITH (MXQSA(I)) GR (0.)
                                        END

LEING FACTORS                                                                    1
LST COMPLETED                                                                    12
IN QUEUE
              AVG QUEUE TIME
    MIN     (IN WORK-HOURS)                                                       1
     *.          **.**
I) MNQSA(I)   AVTAQ(I)

                                        END
```

```
*IBFTC OUT3
               SUBROUTINE OUT3
C       ......GENERATE TABLE 3.
C       ......BEGIN BY WRITING THE HEADING.
               CALL TB3HED
               LET S = STIME
               DO TO 100, FOR EACH PTYPE I, WITH (QTYS(I)) GR (0)
               LET SUM = 0.
   10          DO TO 20, FOR EACH ENTRY OF LIST(I)
C       ......BRING 'CWKNG(ENTRY)' UP TO DATE BEFORE ADDING IT INTO SUM.
               LET C = CWKNG(ENTRY) + WKING(ENTRY) * (S-TWKNG(ENTRY))
               LET CWKNG(ENTRY) = C
               LET SUM = SUM + C
   20          REPEAT 10
               LET UTIL = SUM / CQTY(I)
               CALL TB3LIN (I, UTIL)
               LET CQTY(I) = 0.
C       ......EMPTY OUT EACH 'LIST' SET.
   30          DO TO 50, FOR EACH ENTRY OF LIST(I)
               REMOVE ENTRY FROM LIST(I)
               DESTROY ENTRY
   50          REPEAT 30
  100          LOOP
               RETURN
               END


*IBFTC TB3HED
       REPORT TB3HED
X                                                                          PERSONNEL
X                                              WORK TIME
X                                              AT ACTIVITY
X      PERS      UTIL          MAN-           MAN-            MAN-           MAN-
X      TYPE  QTY FACT   NO    HOURS     NO   HOURS     NO    HOURS     NO   HOURS
                                       END

UTILIZATION                                                                 2


         MAN-         MAN-          MAN-          MAN-          MAN-
  NO    HOURS   NO   HOURS    NO   HOURS    NO   HOURS    NO   HOURS        1
                                       END
```

```
*IBFTC TB3LIN
      REPORT TB3LIN (I, UTIL)
X        ***   **   *.**
 X       I QTYS(I) UTIL
    9          FOR EACH ENTRY CF LIST(I), WITH (FLIST(I)) NE (0)
X                         **    ***.**  **    ***.**  **    ***.**  **    ***.*
 X                         9(ACNC(ENTRY),DECHR(CWKNG(ENTRY)))
                                      END

                                                                             X

                                                                            XX
  **    ***.**  **    ***.**  **    ***.**  **    ***.**  **    ***.**

                                      END


*IBFTC OUT4
              SUBROUTINE OUT4
C       ......GENERATE TABLE 4.
              CALL TB4HED
              LET CUR = CURP
              DO TO 50, FOR EACH ETYPE I, WITH (QTYE(I)) GR (0)
              LET TOTAL = CUR * FLOAT(QTYE(I))
              LET C = CINUS(I)
              LET UTIL = C / TOTAL
              LET FIDLE = TOTAL - C - CDOWN(I)
              CALL TB4LIN (I, FIDLE, UTIL)
              LET NFAIL(I) = 0
              LET CDOWN(I) = 0.
              LET CINUS(I) = 0.
   50         LOOP
              RETURN
              END


*IBFTC TB4HED
      REPORT TB4HED
X                                                                  EQUIPMENT
X                        EQUIPMENT              TIME       IDLE       DOWN
X                           TYPE     QUANTITY  IN USE      TIME       TIME    F
                                      END

TILIZATION                                                           1
O. OF  UTILIZATION  ACTIVITY NOS.
ILURES    FACTOR     WHERE USED                                      1
                                      END
```

```
*IBFTC TB4LIN
      REPORT TB4LIN (I, FIDLE, UTIL)
X                              **          **       **.**      **.**      **.**
 X                              I        GTYE(I)  CINUS(I)   FICLE     CDOWN(I)
   12         FOR EACH ITEM OF SET(I)
X
 X
                                    END

  *       *.***                                                            X
NFAIL(I)  LTIL
                                                                           XX
                       **  **  **  **  **  **  **  **  **  **  **  **
                       12(ACNO(ITEM))
                                    END



*IBFTC OUTS
               SUBROUTINE OUTS
C      ......GENERATE TABLE 5.
               LET C = CURPD
               DO TO 50, FOR EACH SPTYP I
               ACCUMULATE QSP(I) INTO CCSP(I) SINCE TCSP(I)
               LET AVGSP(I) = CGSP(I) / C
               IF (FILL(I)) EC (0), GC TC 50
               LET AVTSQ(I) = DECHR(TISQS(I)) / FLCAT(FILL(I))
   5C          LOOP
               CALL TAB5
               DO TC 100, FCR EACH SPTYP I
               LET CQSP(I) = 0.
               LET TISQS(I) = C.
               LET FQSP = QSP(I)
               LET MXCSP(I) = FCSP
               LET MNCSP(I) = FCSP
               LET DMAND(I) = C
               LET FILL(I) = 0
  ICC          LOOP
               RETURN
               END
```

```
*IBFTC TABS
      REPORT TABS
*
*                                                                         QUEUEING FACTORS BY
*                                                                      REPARABLE QUEUE LENGTH
*
*                                                   COMPONENT                   DEMANDS
*                                                     TYPE       QUANTITY     THIS PERIOD
*                                                       *           **             **
 *                                                      I       QTYSP(I)      DMAND(I) AVQS(
  *   FOR EACH SPTYP I, WITH (QTYSP(I)) GR (0)
                                            END

COMPONENT SPARES TYPE                                                                  12
 AND TIMES BY COMPONENTS                                                               11
 NO. OF REPS IN QUEUE
                                       AVG QUEUE TIME
AVG       MAX       MIN               (IN WORK-HOURS)                                   1
*.**      **.        *.                 **.**
(I)    MXQSP(I)   MNQSP(I)             AVTSC(I)

                                            END


*IBFTC ERROR
             SUBROUTINE ERROR (NAME)
             CALL ERREPT (NAME)
             CALL SNAP (4HERROR)
             STOP
             END


*IBFTC ERREPT
      REPORT ERREPT (NAME)
*     JOB TERMINATED AT TIME    **.*** BECAUSE OF ERROR IN SUBROUTINE **
 *                                RTIME
*                                                                         N
                                                                       (ABRR
                                            END

A**
ME
VIATION)
                                            END


*IBFTC SNAP
             SUBROUTINE SNAP (LABEL)
             CALL SNP1 (LABEL)
             CALL SNP2 (I), FOR EACH SHOP I
             RETURN
             END
```

```
*IBFTC SNP1
      REPORT SNP1 (LABEL)
 X
  X                                                                    SNAPSHCT REQU
 X          RTIME          STIME          PTIME          ENCSH          ENCPC          C
 X           *.*****        *.*****        *.*****        *.*****        *.*****
  X         RTIME          STIME          PTIME          ENCSH          ENCPC          C
 X    SHOP - RIN ROUT TRIN TRCUT MAXR MINR RIP   RIPS      RIPSC          RT
 X          *     *     *     *     *     *     *     *     *.*        **.*
  X         I RIN(I) ROUT(I) TRIN(I) TRCUT(I) MAXR(I) MINR(I) RIP(I) RIPS(
  X
   X   FOR EACH SHOP I
 X     ACTIVITIES ---         QSZA                CCSZA              TCSZA
 X              *              *.*                  *.*              *.*****
  X             I             QSZA(I)             CCSZA(I)           TCSZA(I)
   X   FOR EACH ACTIV I, WITH (MXQSA(I)) NE (0.)
 X     EQUIPMENT TYPES ---       NFAIL         INUSE          CINUS          TINUS
 X              *                 *              *.*           *.*           *.****
  X             I               NFAIL(I)       INUSE(I)       CINUS(I)       TINUS
   X   FOR EACH ETYPE I, WITH (CTYE(I)) NE (0)
 X     SPARE PART TYPES ---      DMAND          FILL           QSP            CQS
 X              *                 *              *             *.*            *.
  X             I               DMAND(I)       FILL(I)        QSP(I)         CQS
   X   FOR EACH SPTYP I, WITH (CTYSP(I)) NE (0)
 X
                                 END

STED AT **A*
          LABEL                                                         2
RPD         CURP          ETIME          CURSH    CURAC    CURSP
.*****      *.*****        *.*****          *        *        *
RPC         CURP          ETIME          CURSH    CURAC    CURSP        2
S           RTSQ       TRTS         TRTSC        MXRT     MNRT   TMXRT   TMNRT
.**         ***.**      **.**       ****.**      *.**     *.**    *.**    *.**
) RIPSC(I) RTS(I) RTSQ(I) TRTS(I) TRTSC(I) MXRT(I) MNRT(I)
                                               TMXRT(I) TMNRT(I)

MXQSA        MNQSA              TIACS         ACCUT                     X
  *.*          *.*               *.**            *
MXQSA(I)     MNQSA(I)           TIACS(I)      ACCUT(I)

       DOWN         CDOWN        TOCWA
        *.*          *.*        *.*****                                 X
(I)    DOWN(I)      CDOWN(I)     TDOWN(I)

      TQSP          MXQSP         MNQSP              TISCS
    *.*****          *.*           *.*               *.**               X
(I)   TQSP(I)       MXQSP(I)      MNQSP(I)           TISCS(I)

                               END                                      X
```

```
*IBFTC SNP2
      REPORT SNP2 (ISHOP)
   14 FOR EACH REP IN LOAD(ISHOP)
X                    REPS IN DEPOT *
 X                                  ISHOP
X      IDNO      *****     *****    *****     *****     *****     *****     ***
 X               14(IDNO(REP))
X      BTIME     *.***     *.***    *.***     *.***     *.***     *.***     *.*
 X               14(BTIME(REP))
X      QTIME     *.***     *.***    *.***     *.***     *.***     *.***     *.*
 X               14(QTIME(REP))
X      QFLAG       *         *        *         *         *         *
 X               14(QFLAG(REP))
                                   END

                                                                        X
                                                                     1
*     *****     *****     *****     *****     *****     *****     *****

*     *.***     *.***     *.***     *.***     *.***     *.***     *.***

*     *.***     *.***     *.***     *.***     *.***     *.***     *.***

*         *         *         *         *         *         *         *   2

                                  END

**********        END DEPOT CAPABILITY OUTPUT PROGRAM                **** 
```

REFERENCES

1. Voosen, B. J., PLANET: Planned Logistics Analysis and Evaluation Technique, The RAND Corporation, RM-4950-PR, January 1967.
2. Voosen, B. J., and David Goldman, PLANET: Part I--Availability and Base Cadre Simulator, The RAND Corporation, RM-4659-PR, April 1967.
3. Voosen, B. J., and M. H. Buchanan, PLANET: Part II--Bench Repair Simulator, The RAND Corporation, RM-4660-PR, April 1967.
4. Voosen, B. J., and Steve Glaseman, PLANET: Part III--Depot Transportation Simulator, The RAND Corporation, RM-4661-PR, April 1967.
5. Voosen, B. J., M. H. Buchanan, and Steve Glaseman, PLANET: Part IV--Depot Repair and Overhaul Simulator, The RAND Corporation, RM-4662-PR, July 1968.

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY

THE RAND CORPORATION

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

PLANET: PART V--REPORTS AND ANALYSIS LIBRARY

4. AUTHOR(S) (Last name, first name, initial)

Voosen, B. J., S. Glaseman, R. J. Young, Judy Judd

5. REPORT DATE

January 1969

6a. TOTAL No. OF PAGES

303

6b. No. OF REFS.

5

7. CONTRACT OR GRANT No.

F44620-67-C-0045

8. ORIGINATOR'S REPORT No.

RM-4663-PR

9a. AVAILABILITY / LIMITATION NOTICES

DDC-1

9b. SPONSORING AGENCY

United States Air Force
Project RAND

10. ABSTRACT

A user's and programmer's manual for the 12 programs comprising the Reports and Analysis Library of PLANET (Planned Logistics Analysis and Evaluation Technique), a logistics prediction and estimating tool designed to help the manager of a system to understand its operation and to find a rationale for allocating resources efficiently. PLANET consists of four computer models that simulate Air Force logistics systems in a single or multibase environment. ~~Whether the models are used singly or in various configurations, the~~ output will be a tape listing of selected variables accumulated during the simulation. From this tape the desired reports are generated by using the Library of programs. The manager can select those programs best suited for analysis of his particular problem. Although PLANET is programmed in SIMSCRIPT, the user need not be a skilled programmer to conduct a simulation. Step-by-step instructions are included to permit the manager to assemble the data in a form ~~acceptable to the simulations~~. Part 1 of the Memorandum contains a brief description of each of the reports and the SIMSCRIPT instructions needed to initialize any of the report programs. Part 2 is the library of programs, including the initialization requirements, a program description oriented to the skilled SIMSCRIPT programmer, and a listing of the SIMSCRIPT SOURCE program.

11. KEY WORDS

PLANET (Maintenance Simulator Model)
Bases
Depots
Maintenance
Logistics
Resource Management
Weapon systems
Computer simulation
Computer programs